

$$\text{pesos} + \text{input} + \text{bias} = z_{n \times 1}$$

$$W_{n \times m} \times m \times p + b_{n \times p} = z_{n \times p} \quad (3.9)$$

Usando un batch para entregar con p ejemplos a la vez:

$$W_{n \times m} \times m \times p + [b_{n \times 1} \dots b_{n \times p}] \xrightarrow{\text{Broadcast}} = z_{n \times p}$$

$$W_{n \times m} \times m \times p + b_{n \times p} = z_{n \times p}$$

batch \uparrow p predictions a la vez

Función de activación softmax a la salida: \rightarrow probabilidad

$$\rightarrow \hat{y}_k = P(Y=k) = \frac{e^{z_{ki}}}{\sum_j e^{z_{ji}}} \quad (4.1)$$

$\rightarrow k = 1, \dots, n$

$\rightarrow j = 1, \dots, n$

Usando batch: (4.2)

$$\rightarrow \hat{y}_{ki} = P(Y=k | X=X_i) = \frac{e^{z_{ki}}}{\sum_j e^{z_{ji}}} \quad (4.2)$$

$\rightarrow k = 1, \dots, n$

$\rightarrow i = 1, \dots, p \quad \rightarrow j = 1, \dots, n$

$$\rightarrow \hat{y} = \frac{e^{z_k}}{\sum_j e^{z_j}}$$

$$\rightarrow \sum_k \hat{y}_{ki} = \frac{\sum_k e^{z_{ki}}}{\sum_j e^{z_{ji}}} = 1 \quad \checkmark$$

\hat{y}_{ki} más alto = predicción del ejemplo i .

Loss Function para el ejemplo i = cross entropy = xentropy

$$L_i = - \sum_k y_{ki} \ln(\hat{y}_{ki}) \quad (5.1)$$

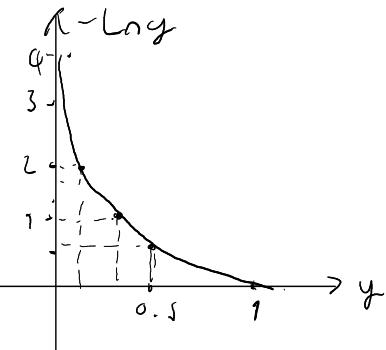
↓
valor correcto
de la salida
 y_{ki} son los elementos de un
one-hot vector!

$$y_{ki} = 0, 0, \dots, 0, 1, 0, \dots, 0 \quad \begin{matrix} \uparrow \\ k \text{ element} \\ \downarrow \\ \text{clase aspirada} \\ (\text{etiqueta}) \end{matrix}$$

$$y_{ki} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} \leftarrow k \text{ element} \\ \leftarrow n \text{ element} \end{matrix}$$

$$\rightarrow L_i = - \ln(\hat{y}_{ki})$$

$$= - \ln \left(\frac{e^{z_{ki}}}{\sum_j e^{z_{ji}}} \right) \quad (5.2)$$



Cost Function para los pesos, bias del batch:

$$J(W, b) = \frac{1}{p} \sum_i L_i \quad (5.3)$$

$$J(W, b) = \frac{1}{p} \sum_i - \ln \left(\frac{e^{z_{ki}}}{\sum_j e^{z_{ji}}} \right)$$

$\rightarrow i = 1, \dots, p$

Para una sola neurona, una sola entrada:

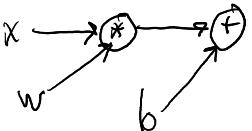
$$x \rightarrow 0 \xrightarrow{w} 0 \rightarrow z$$

$$w x + b = z \quad (6.1)$$

$$\downarrow$$

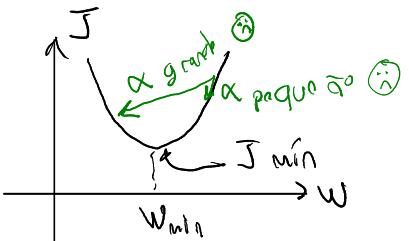
$$J(w, b).$$

Gráfica computacional: Cada operación es un nudo



Gradient descent:

$$\rightarrow \frac{\partial J}{\partial w} = \lim_{h \rightarrow 0} \frac{J(w+h) - J(w)}{h} \quad (6.2)$$



$$w = w - \alpha \frac{\partial J}{\partial w} \quad (6.3)$$

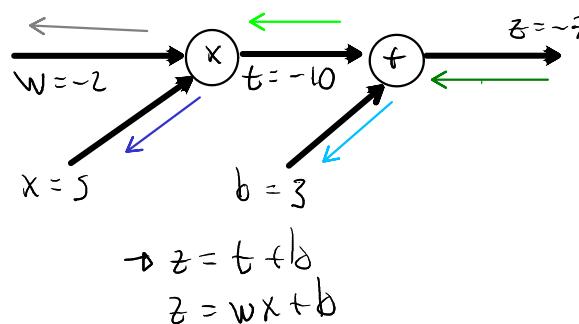
Learning rate
= stop size

Diferencial para b :

$$b = b - \alpha \frac{\partial J}{\partial b} \quad (6.4)$$

Backpropagation:

Gráfica computacional:



Como es la última (y única) capa de la NN:

$$\frac{dz}{dz} = 1 \quad (7.1)$$

$$\rightarrow \frac{dz}{db} = \frac{dz}{dt} \frac{dt}{db} = 1$$

$$\rightarrow \frac{dz}{dt} = \frac{dz}{dt} \frac{dt}{dx} = 1$$

$$\rightarrow \frac{dt}{dx} = \frac{dt}{dx} \frac{dw}{dx} = w = -2 \quad (7.2)$$

$$\rightarrow \frac{dz}{dw} = \frac{dz}{dt} \frac{dt}{dw} = x = 5 \quad (7.3)$$

$$\rightarrow J_m: h = 0.1$$

campos de w

$$\rightarrow w = w_0 + h = -2 + 0.1 = -1.9$$

$$\rightarrow t = w x_0 = (-1.9)(5) = -9.5$$

$$\rightarrow z = t + b_0 = (-9.5) + (3) = -6.5$$

$$\rightarrow z = z_0 + h \frac{dz}{dw}$$

$$z = -7 + (0.1) 5$$

$$z = -6.5 \checkmark$$

$$E_{JM}: h = 0.1$$

↑

Cambiar de X

$$\rightarrow X = X_0 + h = 5 + 0.1 = 5.1$$

$$\rightarrow t = w_0 X = (-2)(5.1) = -10.2$$

$$\rightarrow z = t + b_0 = (-10.2) + (3) = -7.2$$

$$z = z_0 + h \frac{dz}{dx}$$

$$z = -7 + (0.1)(-2)$$

$$z = -7.2 \checkmark$$

$$E_{JM}: h = 0.1$$

↑

Cambiar de b !

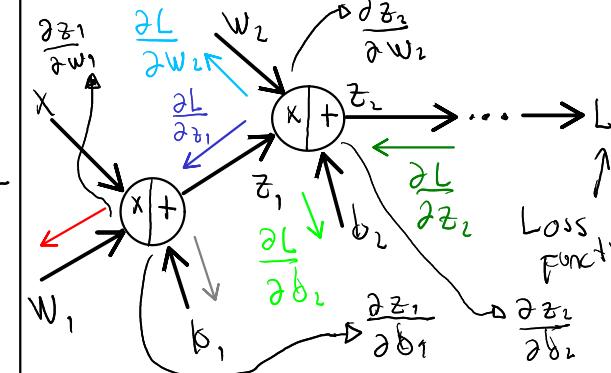
$$\rightarrow b = b_0 + h = 3 + (0.1) = 3.1$$

$$\rightarrow z = t_0 + b = (-10) + (3.1) = -6.9$$

$$z = z_0 + h \frac{dz}{db}$$

$$z = -7 + (0.1)(1) = -6.9 \checkmark$$

Para dos perceptrones:
Gráfica computacional



$$\frac{\partial L}{\partial w_L} = ? \quad \frac{\partial L}{\partial b_i} = ? \quad \frac{\partial L}{\partial w_i} = ? \quad \frac{\partial L}{\partial b_j} = ?$$

$$\rightarrow \frac{\partial L}{\partial b_i} = \frac{\partial L}{\partial z_i} \frac{\partial z_i}{\partial b_i} \rightarrow b_i = b_i - \alpha \frac{\partial L}{\partial b_i}$$

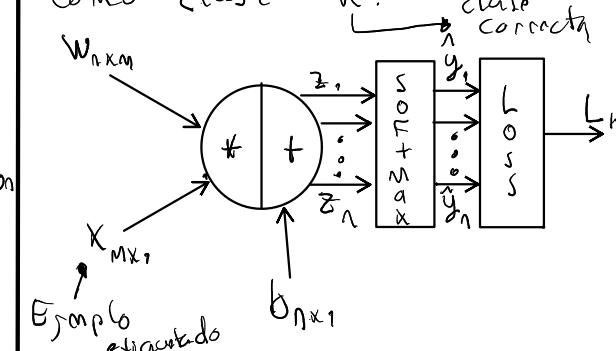
$$\rightarrow \frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial z_i} \frac{\partial z_i}{\partial w_i} \rightarrow w_i = w_i - \alpha \frac{\partial L}{\partial w_i}$$

$$\rightarrow \frac{\partial L}{\partial z_i} = \frac{\partial L}{\partial z_1} \frac{\partial z_1}{\partial z_i}$$

$$\rightarrow \frac{\partial L}{\partial b_i} = \frac{\partial L}{\partial z_1} \frac{\partial z_1}{\partial b_i} \rightarrow b_i = b_i - \alpha \frac{\partial L}{\partial b_i}$$

$$\rightarrow \frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial z_i} \frac{\partial z_i}{\partial w_i} \rightarrow w_i = w_i - \alpha \frac{\partial L}{\partial w_i}$$

Para un solo ejemplo etiquetado como clase K :



$$W_{NKM} X_{MX} + b_{N \times 1} = z_{N \times 1}$$

scores

$$L = - \sum_k y_k \ln \hat{y}_k \quad (S.1)$$

(0 0 ... 1 ... 0)

K elemento

$$L = - \ln \hat{y}_K = - \ln \left(\frac{e^{z_K}}{\sum_j e^{z_j}} \right) \quad (S.2)$$

De la salida esperada para el i -no

$$L = - \ln \left(\frac{e^{z_K}}{\sum_j e^{z_j}} \right) = \ln \sum_j e^{z_j} - z_K$$

$$\rightarrow \frac{\partial L}{\partial z_i} = ?$$

$$\frac{\partial L}{\partial z_i} = \frac{\partial \ln(\sum_j e^{z_j})}{\partial z_i} - \frac{\partial z_K}{\partial z_i}$$

$$= \frac{1}{\sum_j e^{z_j}} \frac{\partial \sum_j e^{z_j}}{\partial z_i} - \delta_{ik}$$

Delta de Kronecker

$$= \frac{e^{z_i}}{\sum_j e^{z_j}} - \delta_{ik}$$

$$\frac{\partial L}{\partial z_i} = \hat{y}_i - y_K \quad (10.1)$$

→ Reducción para la clase i :
Para el score de la salida
esperada (clase correcta):

$$\frac{\partial L}{\partial z_K} = \hat{y}_K - y_K \quad (10.2)$$

Para inicializar W y b en una NN con pocas capas se puede:

$$\rightarrow W_{n \times m} = \text{np.random.rand}(n, m) * 0.01 \quad (11.1)$$

$$\rightarrow b_{n \times 1} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} = \text{np.zeros}((n, 1)) \quad (11.2)$$

$$\rightarrow z = W @ x + b \quad (11.3)$$

$\frac{\partial L}{\partial z}$ en forma matricial: \leftarrow (10.2)

$$\frac{\partial L}{\partial z_{n \times 1}} = \hat{y}_{n \times 1} - \begin{pmatrix} 0 \\ \vdots \\ i \\ \vdots \\ 0 \end{pmatrix} \quad (11.4)$$

$\leftarrow K$ elementos
etiquetas
 \leftarrow One hot vector $= y_{n \times 1}$ \leftarrow en p(6)

$$\rightarrow \frac{\partial L}{\partial W} = \frac{\partial L}{\partial z} \frac{\partial z}{\partial W} = \frac{\partial L}{\partial z} X$$

$$\frac{\partial L}{\partial W_{n \times m}} = \frac{\partial L}{\partial z_{n \times 1}} \cdot (X_{m \times 1})^T \quad (11.5)$$

$$\begin{aligned} \frac{\partial L}{\partial W_{n \times m}} &= (\hat{y}_{n \times 1} - y_{n \times 1}) \cdot (X_{m \times 1})^T \\ &= (y_{-hat} - y) @ X.T \end{aligned} \quad (11.6)$$

$$\rightarrow \frac{\partial L}{\partial b} = \frac{\partial L}{\partial z} \frac{\partial z}{\partial b} = \frac{\partial L}{\partial z} (1)$$

$$\begin{aligned} \frac{\partial L}{\partial b_{n \times 1}} &= \frac{\partial L}{\partial z_{n \times 1}} \cdot (1)_{n \times 1} = \frac{\partial L}{\partial z_{n \times 1}} \\ &= \hat{y}_{n \times 1} - y_{n \times 1} \quad (11.7) \end{aligned}$$

Después se puede actualizar W y b :

$$\rightarrow W_{n \times m} = W_{n \times m} - \alpha \frac{\partial L}{\partial W_{n \times m}} \quad (11.8)$$

$\leftarrow Bjm: 0.01$

$$\rightarrow b_{n \times 1} = b_{n \times 1} - \alpha \frac{\partial L}{\partial b_{n \times 1}} \quad (11.9)$$

\uparrow (G-4)

For a batch with r examples:

$$W_{n \times m} X_{m \times r} + b_{n \times r} = z_{n \times r}$$

Loss functions of r examples:

$$L_{1 \times r} = L_r = -\ln \left[\frac{(e^{z_{ki}})_{1 \times r}}{\left(\sum e^{z_{ji}} \right)_{1 \times r}} \right]$$

De la salida depende para el tipo

Using Jacobian matrix:

$$\frac{\partial L_r}{\partial z_r} = \begin{pmatrix} \frac{\partial L_1}{\partial z_1} & \cdots & \frac{\partial L_1}{\partial z_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial L_r}{\partial z_1} & \cdots & \frac{\partial L_r}{\partial z_n} \end{pmatrix} = \frac{\partial L}{\partial z_{r \times n}}$$

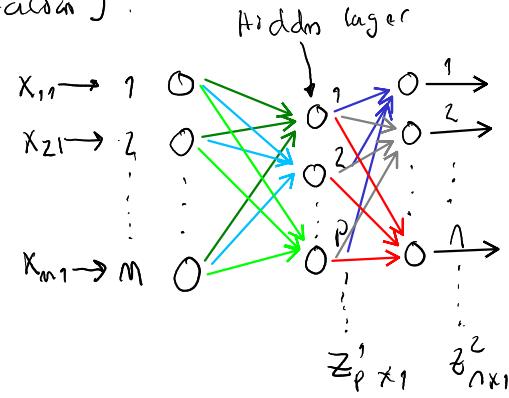
For row i (example i) of r examples:

$$\frac{\partial L}{\partial z_{i \times n}} = \hat{y}_{i \times n} - y_{i \times n}$$

$$\rightarrow \frac{\partial L}{\partial z_{r \times n}} = \hat{y}_{r \times n} - y_{r \times n}$$

\downarrow
One hot vectors:
 $\leftarrow \begin{pmatrix} 0 & 1 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$

Deep learning con dos capas y solo funciones lineales de activación (sin funciones de activación):



$$\rightarrow z'_{p,m} = W^1_{p,m} x_m + b'_{p,m}$$

$$\rightarrow z''_{n',n''} = W^2_{n',n''} z'_{p,m} + b''_{n',n''}$$

$$= W^2_{n',n''} (W^1_{p,m} x_m + b'_{p,m}) + b''_{n',n''}$$

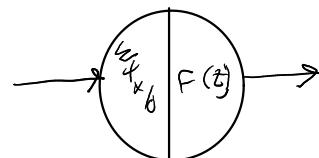
$$= (W^2 W^1)_{n',n''} x_m + (W^2 b^1)_{n',n''} + b''_{n',n''}$$

$$= (W^2 W^1)_{n',n''} x_m + (W^2 b^1 + b^2)_{n',n''} \quad (12.1)$$

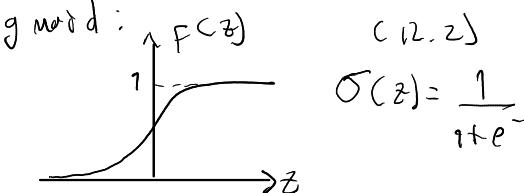
Entonces las hidden layer no actúan, dando igual usar la sola

capa de salida con n perceptrones

Funciones de activación:



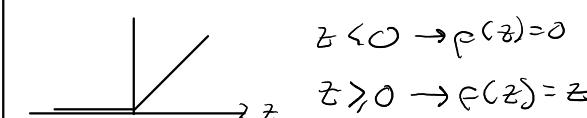
Sigmoid:



$$(12.2)$$

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

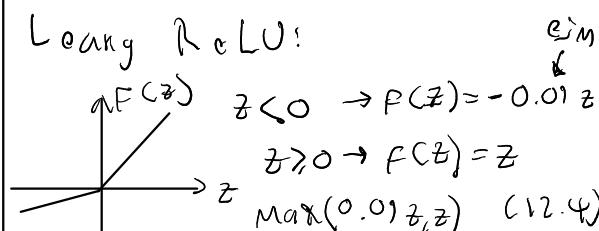
ReLU: Rectified Linear Unit:



$$\max(0, z) \quad (12.3)$$

máximo entre 0 y z

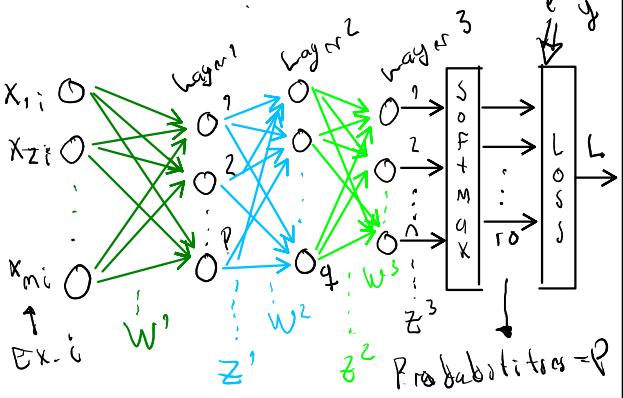
Leaky ReLU:



$$\max(0.01z, z)$$

$$(12.4)$$

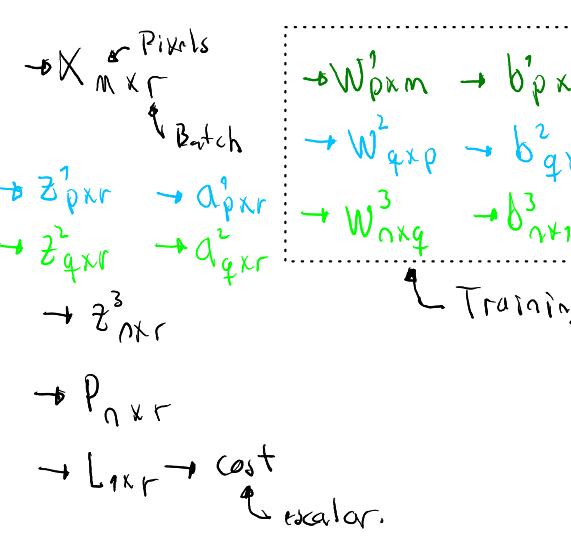
Usando funciones de activación con dos hidden layers:



R: Ejemplos totales

Usando un batch de r ejemplos:

$\frac{R}{r}$ = pasadas para entrenar con los r elementos
 ↪ Forward pass (backpropagation)



Forward pass:

$$z^1_{pxr} = W^1_{pxm} X_{m \times r} + (b^1_{px1} \dots b^1_{pxr})_{pxr}$$

Después de aplicar la función de activación de la capa j :

$$a^1_{pxr} = f^1(z^1_{pxr})$$

$$z^2_{qxr} = W^2_{qxp} a^1_{pxr} + (b^2_{qxr1} \dots b^2_{qxr})_{qxr}$$

$$a^2_{qxr} = f^2(z^2_{qxr})$$

$$\rightarrow z^3_{nxr} = W^3_{nxq} a^2_{qxr} + b^3_{nxr}$$

Softmax:

$$P_{n \times r} = \frac{(e^{z^3})_{n \times r}}{\left(\sum_j e^{z^3_j} \right)_{n \times r}} = \hat{y}_{n \times r}$$

sobre cada columna (c_j) de $(P^{z^2})_{n \times r}$

Cross entropy:

$$\rightarrow L_{1 \times r} = -\sum_i (P_{a_i}, P_{a_2} \dots P_{a_r})_{1 \times r}$$

a_i etiquetas a_{j-2} etiquetas

$$\rightarrow \text{cost} = \frac{\sum_i L_{1 \times r}}{r}$$

Backpropagation:

$$\rightarrow \frac{\partial L}{\partial z^3_{nxr}} = P_{nxr} - y_{nxr}$$

etiquetas

one hot vector por cada ejemplo:

$(0 \dots 0)$	$\leftarrow 1$
$(1 \dots 0)$	$\leftarrow 2$
$(0 \dots 1)$	$\leftarrow n$
$(0 \dots 0)$	$\leftarrow r$

$$\rightarrow \frac{\partial L}{\partial a^2_{qxr}} = \frac{\partial L}{\partial z^3} \frac{\partial z^3}{\partial a^2}$$

$$\frac{\partial L}{\partial a^2_{qxr}} = (W^3_{nxq})^T \frac{\partial L}{\partial z^3_{nxr}}$$

most of formula is red

$$\rightarrow \frac{\partial L}{\partial W^3} = \frac{\partial L}{\partial z^3} \frac{\partial z^3}{\partial W^3}$$

$$\frac{\partial L}{\partial W^3_{nxq}} = \frac{\partial L}{\partial z^3_{nxr}} (a^2_{qxr})^T$$

normalized

$\rightarrow \frac{\partial L}{\partial b^3} = \frac{\partial L}{\partial z^3} \frac{\partial z^3}{\partial b^3}$ I

$\frac{\partial L}{\partial b_{n \times 1}} = \left(\sum_j^r \frac{\partial L}{\partial z_{1j}^3}, \sum_j^r \frac{\partial L}{\partial z_{2j}^3}, \dots, \sum_j^r \frac{\partial L}{\partial z_{nj}^3} \right)_{1 \times r}$ sum elements of each row / r normalized

$\rightarrow \frac{\partial L}{\partial z^2} = \frac{\partial L}{\partial a^2} \frac{\partial a^2}{\partial z^2}$ W^3 normalized

$\frac{\partial L}{\partial z^2}_{q \times r} = \left(\frac{\partial L}{\partial a_{1j}^2} \cdot \frac{\partial F^L(z_{1j}^2)}{\partial z^2}, \frac{\partial L}{\partial a_{2j}^2} \cdot \frac{\partial F^L(z_{2j}^2)}{\partial z^2}, \dots, \frac{\partial L}{\partial a_{nj}^2} \cdot \frac{\partial F^L(z_{nj}^2)}{\partial z^2} \right)_{q \times r}$

$\rightarrow \frac{\partial L}{\partial z^2} = \frac{\partial L}{\partial b^2} \frac{\partial b^2}{\partial z^2}$ W^3 normalized

$\frac{\partial L}{\partial b^2} = \frac{\partial L}{\partial z^2} \frac{\partial z^2}{\partial b^2}$ I

$\rightarrow \frac{\partial L}{\partial a^1} = \frac{\partial L}{\partial z^1} \frac{\partial z^1}{\partial a^1}$

$\frac{\partial L}{\partial a^1}_{p \times r} = (W^2_{q \times p})^T \frac{\partial L}{\partial z^1}_{q \times r}$ most b^1 normalized

$\rightarrow \frac{\partial L}{\partial z^1} = \frac{\partial L}{\partial q^1} \frac{\partial q^1}{\partial z^1}$ W^2 normalized

$\frac{\partial L}{\partial z^1}_{p \times r} = \left(\frac{\partial L}{\partial a_{1j}^1} \frac{\partial F^{-1}(z_{1j}^1)}{\partial z^1}, \frac{\partial L}{\partial a_{2j}^1} \frac{\partial F^{-1}(z_{2j}^1)}{\partial z^1}, \dots, \frac{\partial L}{\partial a_{nj}^1} \frac{\partial F^{-1}(z_{nj}^1)}{\partial z^1} \right)_{p \times r}$

$\rightarrow \frac{\partial L}{\partial z^1} = \frac{\partial L}{\partial b^1} \frac{\partial b^1}{\partial z^1}$ I

$\frac{\partial L}{\partial b^1}_{p \times r} = \left(\sum_j^r \frac{\partial L}{\partial z_{1j}^1}, \sum_j^r \frac{\partial L}{\partial z_{2j}^1}, \dots, \sum_j^r \frac{\partial L}{\partial z_{nj}^1} \right)_{p \times r}$ OK

Training for each epoch, for each batch:

$\rightarrow W'_{p \times m} = W_{p \times n} - \alpha \frac{\partial L}{\partial W^1_{p \times m}}$

$\rightarrow b'_{p \times 1} = b_{p \times 1} - \alpha \frac{\partial L}{\partial b^1_{p \times 1}}$

$\rightarrow W^2_{q \times p} = W^2_{q \times p} - \alpha \frac{\partial L}{\partial W^2_{q \times p}}$

$\rightarrow b^2_{q \times 1} = b^2_{q \times 1} - \alpha \frac{\partial L}{\partial b^2_{q \times 1}}$

$$\rightarrow \mathbf{W}_{n \times q}^3 = \mathbf{W}_{n \times q}^3 - \alpha \frac{\partial L}{\partial \mathbf{W}_{n \times q}^3}$$

$$\rightarrow \mathbf{b}_{n \times 1}^3 = \mathbf{b}_{n \times 1}^3 - \alpha \frac{\partial L}{\partial \mathbf{b}_{n \times 1}^3}$$

Accuracy using validation data:
after each epoch

Forward (Validation Data)

$$\text{softmax}(\mathbf{z}_{n \times r}^3)$$

$$\mathbf{P}_{n \times r}$$

$$\text{Prediction} = \mathbf{P}_c = (\mathbf{P}_{\max, 1}, \dots, \mathbf{P}_{\max, r})_{n \times r}$$

max row col. r

max row of col. 1

$$\text{Accuracy} = \frac{\sum_{i=1}^{\# \text{ batches}} \sum_{j=1}^r \delta p_{e_j} = \text{Label}_j}{\# \text{ Validation data}} \cdot \frac{\# \text{ batches} \cdot r}{\# \text{ batches} \cdot r}$$

Prediction of a image at time t
epochs: →

Optimal

→ $\mathbf{W}_{p \times m}^1 \rightarrow \mathbf{b}_{p \times 1}^1$
 → $\mathbf{W}_{q \times p}^2 \rightarrow \mathbf{b}_{q \times 1}^2$
 → $\mathbf{W}_{n \times q}^3 \rightarrow \mathbf{b}_{n \times 1}^3$

$$\rightarrow \text{image}_{1 \times m} = \mathbf{X}_{1 \times m}$$

→ Total pixels

↓
Forward ($\mathbf{X}_{1 \times m}$)

$$\mathbf{z}_{n \times 1}^3$$

$$\downarrow \text{softmax}$$

$$\mathbf{g}_{n \times 1} > \mathbf{p}_{n \times 1}$$

$$\text{Prediction} = \mathbf{P}_c = \mathbf{P}_{\max} \text{ of } \mathbf{P}_{n \times 1}$$

Usando Matriz Jacobiana:
 $\frac{\partial L_r}{\partial z^3} \rightarrow$ loss función de r ejemplos

$$\frac{\partial L_r}{\partial z^3}$$

$$= \begin{pmatrix} \frac{\partial L_1}{\partial z^3} & \dots & \frac{\partial L_1}{\partial z^n} \\ \vdots & \ddots & \vdots \\ \frac{\partial L_r}{\partial z^3} & \dots & \frac{\partial L_r}{\partial z^n} \end{pmatrix} = \frac{\partial L}{\partial z^{r \times n}}$$

$$\frac{\partial L}{\partial z^3_{r \times n}} = \hat{y}_{r \times n} - y_{r \times n}$$

one hot vectors!

$$\begin{pmatrix} 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}$$

$$\rightarrow \frac{\partial L_r}{\partial a_q^2} = \frac{\partial L_r}{\partial z_n^3} \frac{\partial z_n^3}{\partial a_q^2}$$

$$\frac{\partial L}{\partial a_{r \times q}^2} = \frac{\partial L}{\partial z_n^3} \frac{\partial z_n^3}{\partial a_{n \times q}^2}$$

$$= (\hat{y}_{r \times n} - y_{r \times n}) W_{n \times q}^3$$

$$\rightarrow \frac{\partial L_r}{\partial w_{n \times q}^3} = \frac{\partial L_r}{\partial z_n^3} \frac{\partial z_n^3}{\partial w_{n \times q}^3}$$

$$\frac{\partial L}{\partial w_{r \times n \times q}^3} = (\hat{y}_{r \times n} - y_{r \times n}) q_q^2$$

Products
transmision?

$$\rightarrow \frac{\partial L_r}{\partial b_n^3} = \frac{\partial L_r}{\partial z_n^3} \frac{\partial z_n^3}{\partial b_n^3}$$

$$\frac{\partial L}{\partial b_{r \times n}^3} = \frac{\partial L}{\partial z_{r \times n}^3} I_{n \times n} = \frac{\partial L}{\partial z_{r \times n}^3}$$

$$\rightarrow \frac{\partial L_r}{\partial w_{q \times p}^2} = \frac{\partial L_r}{\partial z_n^3} \frac{\partial z_n^3}{\partial a_q^2} \frac{\partial a_q^2}{\partial z_q^2} \frac{\partial z_q^2}{\partial w_{q \times p}^2}$$

$$= (\hat{y}_{r \times n} - y_{r \times n}) W_{n \times q}^3 \left(\frac{\partial f^2(z^2)}{\partial z^2} \right)_{q \times q}$$

$$= \frac{\partial L}{\partial w_{r \times q \times p}^2} \frac{\partial L_r}{\partial z_q^2}$$

$$\rightarrow \frac{\partial L_r}{\partial b_q^2} = \frac{\partial L_r}{\partial z_n^3} \frac{\partial z_n^3}{\partial a_q^2} \frac{\partial a_q^2}{\partial z_q^2} \frac{\partial z_q^2}{\partial b_q^2}$$

$I_{q \times q}$

$$= \frac{\partial L}{\partial b_{r \times q}^2}$$

$$\rightarrow \frac{\partial L_r}{\partial w_{p \times m}^1} = \frac{\partial L_r}{\partial z_n^3} \frac{\partial z_n^3}{\partial a_q^2} \frac{\partial a_q^2}{\partial z_q^2} \frac{\partial z_q^2}{\partial a_p^1}$$

$$+ \frac{\partial a_p^1}{\partial z_p^1} \frac{\partial z_p^1}{\partial w_{p \times m}^1}$$

$$\frac{\partial L}{\partial w_{r \times p \times m}^1} = \frac{\partial L_r}{\partial z_n^3} W_{q \times p}^2 \left(\frac{\partial f^1(z^1)}{\partial z^1} \right)_{p \times p}$$

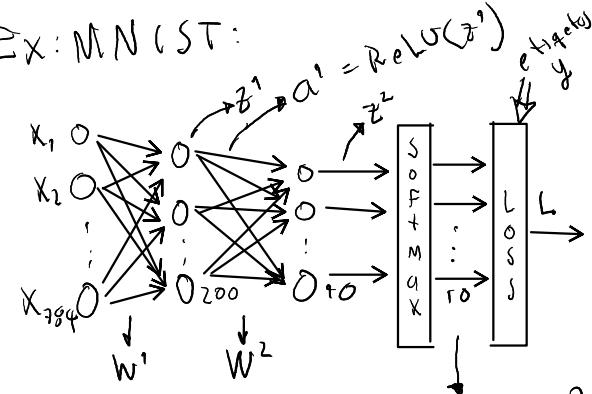
$\frac{\partial L_r}{\partial z_p^1}$

$$\rightarrow \frac{\partial L_r}{\partial b_p^1} = \frac{\partial L_r}{\partial z_p^1} \frac{\partial z_p^1}{\partial b_p^1}$$

$I_{p \times p}$

$$= \frac{\partial L}{\partial b_{r \times p}^1}$$

Ex: MNIST:



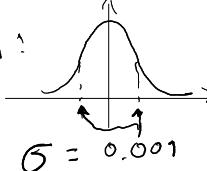
$\rightarrow X_{64 \times 784}$ Pixels
 $\rightarrow W_{200 \times 784}$
 $\rightarrow b_{200 \times 1}$
 $\rightarrow Z_{200 \times 64}$
 $\rightarrow a_{200 \times 64}$
 $\rightarrow z^1_{10 \times 64}$
 $\rightarrow P_{10 \times 64}$
 $\rightarrow L_{1 \times 64} \rightarrow \text{cost}$
 \uparrow escalar.

Inicialmente:

$$W_{200 \times 784} =$$

$$\text{np.random.randn}(200, 784) \cdot 0.001$$

Distribución normal:



$$\rightarrow b_{200 \times 1} = \text{np.zeros}((200, 1))$$

$$\rightarrow W_{10 \times 200} =$$

$$\text{np.random.randn}(10, 200) \cdot 0.001$$

$$\rightarrow b_{10 \times 1} = \text{np.zeros}((10, 1))$$

Forward:

$$z^1_{200 \times 64} = W_{200 \times 784} (X_{64 \times 784})^T + (b^1_1 \ b^1_2 \dots b^1_{10})_{200 \times 64}$$

$$\rightarrow a^1_{200 \times 64} = \text{ReLU}(z^1_{200 \times 64})$$

$$\rightarrow z^2_{10 \times 64} = W_{10 \times 200} a^1_{200 \times 64} + (b^2_1 \ b^2_2 \dots b^2_{10})_{10 \times 64}$$

Softmax:

$$P_{10 \times 64} = \frac{(e^{z^2})_{10 \times 64}}{\left(\sum_{ij} e^{z^2_{ij}} \right)_{10 \times 64}} = g_{10 \times 64} \quad (\text{eq. 1})$$

suma cada columna ($e_{j|m}$)

Cross entropy:

(S.2)

$$(e^L)_{1 \times 64} = (P_{a_1}, P_{a_2} \dots P_{a_{64}})$$

$a_{j,1}$ etiquetas $a_{j,-2}$ etiquetas

$$\rightarrow L_{1 \times 64} = -\ln(e^L)_{1 \times 64}$$

$$\rightarrow \text{cost} = \frac{\sum_{i=1}^{64} L_{1,i}}{64} \quad (\text{S.3})$$

Backward:

$$\rightarrow \frac{\partial L}{\partial z^2_{10 \times 64}} = P_{10 \times 64} - y_{10 \times 64}$$

one hot vector para cada ejemplo:

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \leftarrow 1 \quad \leftarrow 2 \quad \leftarrow 10 \quad \leftarrow 64$$

$$\rightarrow \frac{\partial L}{\partial W^2} = \frac{\partial L}{\partial z^2} \frac{\partial z^2}{\partial W^2}$$

$$\frac{\partial L}{\partial W_{10 \times 200}} = \frac{\partial L}{\partial z^2} \frac{a^1_{200 \times 64}}{10 \times 64} \quad \text{G4}$$

$$\rightarrow \frac{\partial L}{\partial b^2} = \frac{\partial L}{\partial z^2} \frac{\partial z^2}{\partial b^2} \quad \text{I}$$

$$\frac{\partial L}{\partial b^2_{10 \times 1}} = \begin{cases} \frac{\sum_{j=1}^{64} \frac{\partial L}{\partial z^2_{1,j}}}{64} & \text{sum_elements of each row} \\ \frac{\sum_{j=1}^{64} \frac{\partial L}{\partial z^2_{10,j}}}{10} & /64 \end{cases}$$

$$\frac{\partial L}{\partial a^1} = \frac{\partial L}{\partial z^1} \frac{\partial z^1}{\partial a^1}$$

most normalized

$$\frac{\partial L}{\partial a_{200 \times 64}} = (W^1_{10 \times 200})^T \frac{\partial L}{\partial z^1_{10 \times 64}}$$

$$\frac{\partial L}{\partial z^1} = \frac{\partial L}{\partial a^1} \frac{\partial a^1}{\partial z^1}$$

normalized

$$\frac{\partial L}{\partial z^1_{200 \times 64}} = \left(\frac{\partial L}{\partial a^1_{ij}} \delta_{z^1_{ij} > 0} \right)_{200 \times 64}$$

element ij

When $z^1_{ij} > 0$:
 $\delta_{z^1_{ij} > 0} = 1$
 $z^1_{200 \times 64}$

When $z^1_{ij} \leq 0$:
 $\delta_{z^1_{ij} > 0} = 0$

$$\frac{\partial L}{\partial W^1} = \frac{\partial L}{\partial z^1} \frac{\partial z^1}{\partial W^1}$$

$$\frac{\partial L}{\partial W^1_{200 \times 784}} = \frac{\partial L}{\partial z^1_{200 \times 784}} X_{64 \times 784}$$

OK

$$W^2 = \frac{\partial z^2}{\partial a^1} \rightarrow \frac{\partial L}{\partial a^1}$$

$$\frac{\partial L}{\partial b^1} = \frac{\partial L}{\partial z^1} \frac{\partial z^1}{\partial b^1}$$

$$\frac{\partial L}{\partial b^1_{200 \times 1}} = \frac{\partial L}{\partial z^1_{200 \times 1}} - \frac{\partial L}{\partial b^1_{200 \times 1}}$$

$$W^2_{10 \times 200} = W^2_{10 \times 200} - \frac{\partial L}{\partial W^2_{10 \times 200}}$$

$$b^2_{10 \times 1} = b^2_{10 \times 1} - \frac{\partial L}{\partial b^2_{10 \times 1}}$$

OK

$$\frac{\partial L}{\partial b^1_{200 \times 1}} = \left(\sum_j \frac{\partial L}{\partial z^1_{1j}} \right)_{200 \times 1}$$

Training for each epoch, for each batch:

$$\alpha = 0.001$$

$$W^1_{200 \times 784} = W^1_{200 \times 784} - \frac{\partial L}{\partial W^1_{200 \times 784}}$$

$$b^1_{200 \times 1} = b^1_{200 \times 1} - \frac{\partial L}{\partial b^1_{200 \times 1}}$$

$$W^2_{10 \times 200} = W^2_{10 \times 200} - \frac{\partial L}{\partial W^2_{10 \times 200}}$$

$$b^2_{10 \times 1} = b^2_{10 \times 1} - \frac{\partial L}{\partial b^2_{10 \times 1}}$$

Accuracy using validation data after each epoch:

Forward (Validation Data)

Softmax ($z^2_{10 \times 64}$)

$P_{10 \times 64}$:

Prediction = $P_e = (P_{\max,1}, \dots, P_{\max,64})_{10 \times 64}$

Max row of P_e ← Max row col. 64

Validation data.

$$\text{Accuracy} = \frac{\# \text{ batches}}{\# \text{ Validation data}} \sum_{i=1}^{64} \sum_{j=1}^{10} \delta_{p_{e,j} = l_{\text{label},j}}$$

Prediction of a image at trn

epochs: →

Optimal {

- $W^1_{200 \times 784}$
- $b^1_{200 \times 1}$
- $W^2_{10 \times 200}$
- $b^2_{10 \times 1}$

→ image $z^1_{1 \times 784} = X_{1 \times 784}$

→ $z^1_{200 \times 1} = W^1_{200 \times 784} (X_{1 \times 784})^T + b^1_{200 \times 1}$

→ $a^1_{200 \times 1} = \text{ReLU}(z^1_{200 \times 1})$

→ $z^2_{10 \times 1} = W^2_{10 \times 200} a^1_{200 \times 1} + b^2_{10 \times 1}$

→ $P_{10 \times 1} = \frac{(e^{z^2})_{10 \times 1}}{\sum_{j=1}^{10} e^{z^2_{j1}}} = \hat{y}_{10 \times 1}$

Prediction = $\hat{P}_e = \text{P}_{\max} \text{ OF } P_{10 \times 1}$

Min. value of pixels in MNIST:

$$\min = 0 \leftarrow \text{Black}$$

Max value:

$$\max = 255 \leftarrow \text{White}$$

Normalize pixels to mean=0, std=1:

\rightarrow P numbers
of every pixels
of every sample

mean, std

of each pixel} data

$$\rightarrow \text{Norm}_{S \times P} = \frac{\text{data}_{S \times P} - \text{mean}}{\text{std}} \quad (17.3)$$

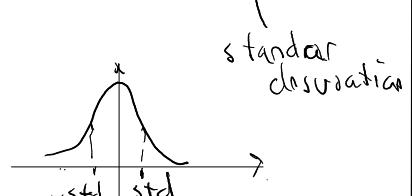
$$\rightarrow \text{Norm. mean}() = 0 \quad (17.4)$$

$$\rightarrow \text{Norm. std}() = 1$$

Matrix $N \times M$ w rnd #s by normal distribution, mean θ : (17.1)

$$W_{N \times M} = \text{randn}(n, m) * \text{std}$$

$$\begin{matrix} 0 & \xrightarrow{\text{P}} \\ 0 & \xrightarrow{\text{P}} \\ \vdots & \vdots \\ 0_m & \xrightarrow{\text{P}} \end{matrix}$$



Xavier He init: (17.5)

$$W_{N \times M} = \text{randn}(n, m) \left(\frac{1}{\sqrt{M}} \right)$$

Kaiming init: (17.6)

$$W_{N \times M} = \text{randn}(n, m) \left(\frac{1}{\sqrt{M/2}} \right)$$

Vanishing gradient problem!
Forward:

$$z^1 = W^1 x + b^1$$

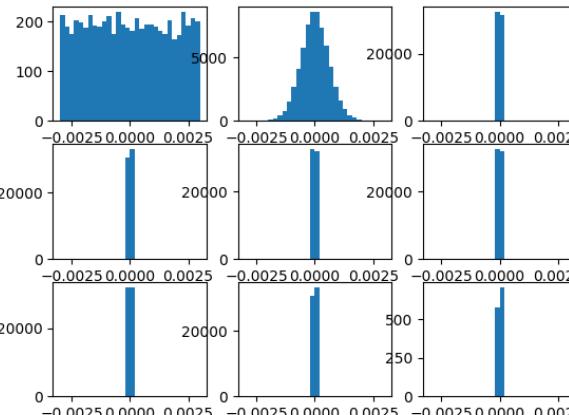
$$z^2 = W^2 f(z^1) + b^2$$

$$z^i = W^i f(z^{i-1}) + b^i$$

For normal init:

$$\begin{aligned} z^2 &< z^1 \\ \vdots & \\ z^i &< z^{i-1} \end{aligned} \quad (17.7)$$

Histograms to z^1, \dots, z^i :



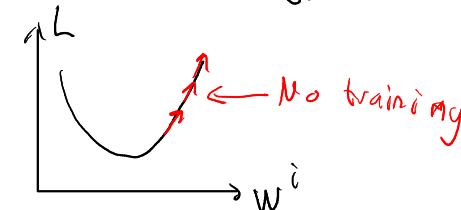
So, Backward!

$$\rightarrow \frac{\partial L}{\partial W^i} = \frac{\partial L}{\partial z^i} \underbrace{\frac{\partial z^i}{\partial W^i}}_{\text{Fix}} \quad (17.8)$$

$$\rightarrow \frac{\partial L}{\partial W^{i-1}} = \frac{\partial L}{\partial z^{i-1}} \underbrace{\frac{\partial z^{i-1}}{\partial W^{i-1}}}_{\text{Fix}} \quad (17.8)$$

Training:

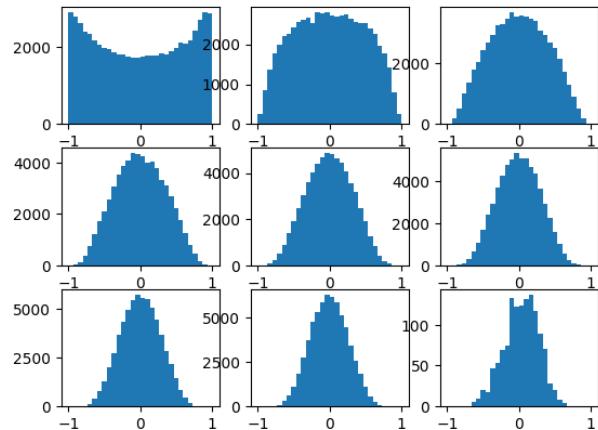
$$\rightarrow W^i = W^i - \alpha \frac{\partial L}{\partial W^i} \quad \text{Fix} \quad (17.9)$$



Ex. of No Vanishing problem
using Xavier, $f = \tanh$ and MNIST

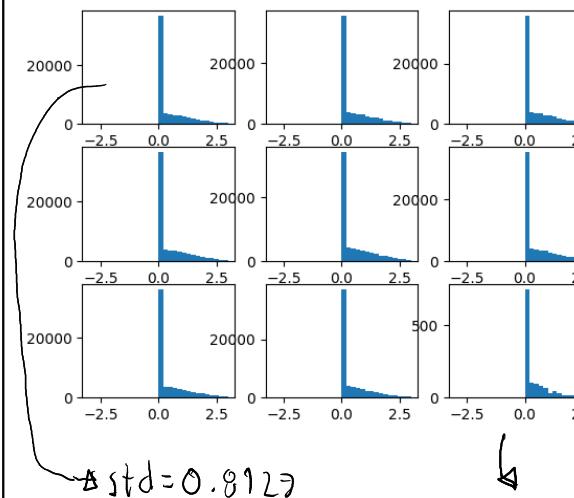
$$\xrightarrow{\text{X.}}$$

Histogram of z^1, \dots, z^g :



Ex. of No Vanishing problem
using Kaiming, $f = \text{ReLU}$, MNIST:

Histogram of z^1, \dots, z^g :



$$\text{std} = 0.894$$

No Vanishing

Tips to avoid Vanishing:

Normal \rightarrow Nonp $p(z)$

Xavier $\rightarrow \tanh(z)$

Kaiming He $\rightarrow \text{ReLU}(z)$
 $\rightarrow \tanh(z)$

