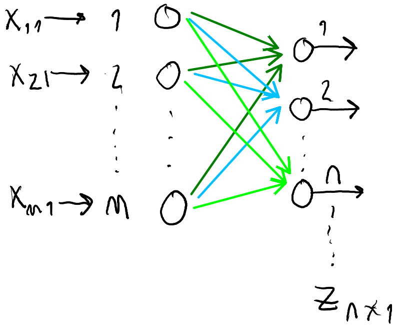


pesos * input + biases $= z_{n\times 1}$

$$W_{n\times m} X_{m\times 1} + b_{n\times 1} = z_{n\times 1} \quad (3.1)$$

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

$$W_{n\times m} X_{m\times p} + \underbrace{[\overset{1}{b_{n\times 1}} \cdots \overset{p}{b_{n\times 1}}]}_{\text{Broadcast}} = z_{n\times p}$$

$$W_{n\times m} \underset{\text{batch}}{I_{m\times p}} + b_{n\times p} = \underset{\text{p predictions a la vez}}{z_{n\times p}}$$

Función de activación softmax a la salida: → probabilidad

$$\hat{y}_k = P(Y=k) = \frac{e^{z_{k1}}}{\sum_j e^{z_{j1}}} \quad (4.1)$$

→ $k = 1, \ldots, n$

→ $j = 1, \ldots, n$

Usando batch: (4.2)

$$\hat{y}_{ki} = P(Y=k \mid X=x_i) = \frac{e^{z_{ki}}}{\sum_j e^{z_{ji}}}$$

→ $k = 1, \ldots, n$

→ $i = 1, \ldots, p$

→ $j = 1, \ldots, n$

$$\hat{y} = \frac{e^{s_k}}{\sum_j e^{s_j}}$$

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

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

Loss Function para el ejemplo i = Cross entropy = Xentropy

$$L_i = -\sum_k \underset{\text{valor correcto de la salida}}{y_{ki}} \ln(\hat{y}_{ki}) \quad (5.1)$$

y_{ki} son los elementos de un one-hot vector:

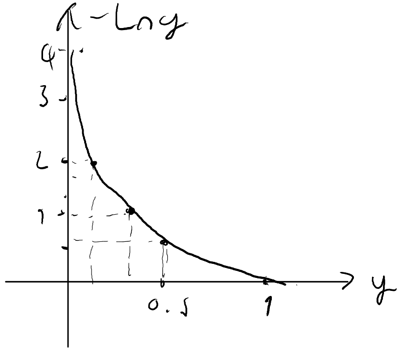
$$y_{ki} = 0, 0, \ldots, 0, \underset{\substack{k \text{ element} \\ \downarrow \\ \text{clase esperada (etiqueta)}}}{1}, 0, \ldots, \overset{n}{0}$$

$$y_{ki} = \begin{array}{cl} \left[\begin{array}{c} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{array}\right] & \begin{array}{l} \\ \\ \\ \\ \leftarrow k \text{ element} \\ \\ \\ \leftarrow n \text{ element} \end{array} \end{array}$$

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

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

De la salida esperada para el ej. i



Cost Function para los pesos, biases del batch:

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

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

→ $i = 1, \ldots, p$

Para una sola neurona, una sola entrada:

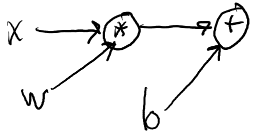
$$x \to \circ \xrightarrow{w} \circ \to z$$

$$wx + b = z \qquad (6.1)$$

$$\downarrow$$

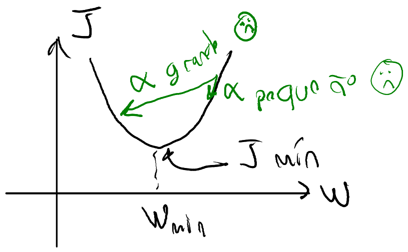
$$J(w, b)$$

Gráfica computacional: Cada operación es un nodo



Gradient descent:

$$\to \frac{\partial J}{\partial w} = \lim \frac{J(w+h) - J(w)}{h} \qquad (6.2)$$



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

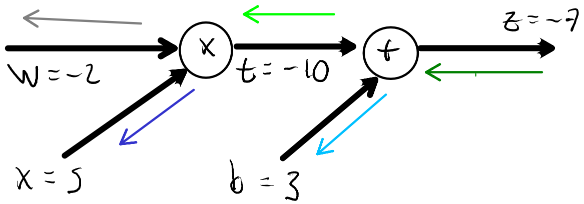
α → learning rate = step size

De igual manera para b:

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

Backpropagation:

Gráfica computacional:



$$\to z = t + b$$

$$z = wx + b$$

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

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

$$\to \frac{dz}{db} = \frac{dz}{dz} \frac{dz}{db} = 1$$

$$\to \frac{dz}{dt} = \frac{dz}{dz} \frac{dz}{dt} = 1$$

$$\to \frac{dz}{dx} = \frac{dz}{dt} \frac{dt}{dx} = w = -2 \qquad (7.2)$$

$$\to \frac{dz}{dw} = \frac{dz}{dt} \frac{dt}{dw} = x = 5 \qquad (7.3)$$

Ejm: $h = 0.1$

cambio de w

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

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

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

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

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

$$z = -6.5$$

Ejm: $h = 0.1$

↑ cambio de x

$\rightarrow x = x_0 + h = 5 + 0.1 = 5.1$

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

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

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

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

$$z = -7.2 \checkmark$$

Ejm: $h = 0.1$

↑ cambio de b:

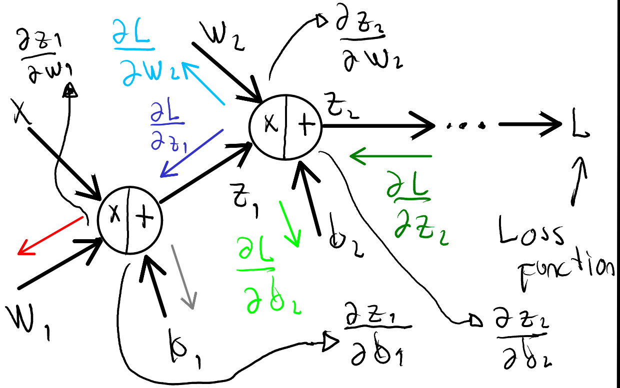
$\rightarrow b = b_0 + h = 3 + (0.1) = 3.1$

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

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

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

Para dos perceptrons: Gráfica computacional



$$\frac{\partial L}{\partial w_2} = ? \quad \frac{\partial L}{\partial b_2} = ? \quad \frac{\partial L}{\partial w_1} = ? \quad \frac{\partial L}{\partial b_1} = ?$$

$$\rightarrow \frac{\partial L}{\partial b_2} = \frac{\partial L}{\partial z_2}\frac{\partial z_2}{\partial b_2} \qquad \rightarrow b_2 = b_2 - \alpha \frac{\partial L}{\partial b_2}$$

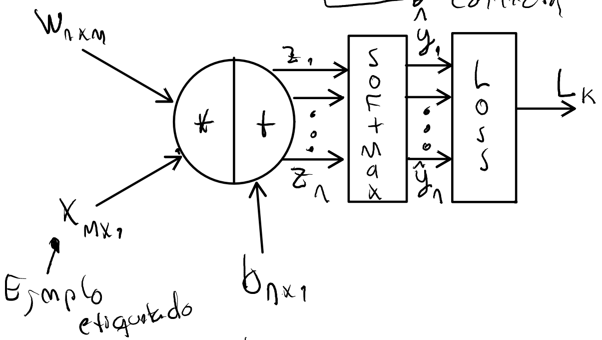
$$\rightarrow \frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial z_2}\frac{\partial z_2}{\partial w_2} \qquad \rightarrow w_2 = w_2 - \alpha \frac{\partial L}{\partial w_2}$$

$$\rightarrow \frac{\partial L}{\partial z_1} = \frac{\partial L}{\partial z_2}\frac{\partial z_2}{\partial z_1}$$

$$\rightarrow \frac{\partial L}{\partial b_1} = \frac{\partial L}{\partial z_1}\frac{\partial z_1}{\partial b_1} \qquad \rightarrow b_1 = b_1 - \alpha \frac{\partial L}{\partial b_1}$$

$$\rightarrow \frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial z_1}\frac{\partial z_1}{\partial w_1} \qquad \rightarrow w_1 = w_1 - \alpha \frac{\partial L}{\partial w_1}$$

Para un solo ejemplo etiquetado como clase k: → clase correcta



$$W_{nxm} X_{mx1} + b_{nx1} = z_{nx1}$$

↑ scores

$$\rightarrow L = -\sum_k y_k \, Ln \, \hat{y}_k \leftarrow (5.1)$$

$(0\ 0 \ldots 1 \ldots 0)$ ↑ k elemento

$$L = -Ln \, \hat{y}_k = -Ln\left(\frac{e^{z_k}}{\sum_j e^{z_j}}\right) \quad (5.2)$$

De la salida esperada para el ejemplo

$j = 1, \ldots, n$

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

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

$$\frac{\partial L}{\partial z_i} = \frac{\partial Ln\left(\sum_j e^{z_j}\right)}{\partial z_i} - \frac{\partial z_k}{\partial z_i}$$

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

↑ Delta de Kronecker

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

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

→ Predicción 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:

→ $W_{n\times m}$ = np.random.randn$\underbrace{(n,m)}_{n\times m}$*0.01 (11.1)

→ $b_{n\times1} = \begin{pmatrix}0\\ \vdots \\ 0\end{pmatrix}$ = np.zeros((n,1)) (11.2)

→ $z = W@x + b$ (11.3)

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

$$\frac{\partial L}{\partial z_{n\times1}} = \hat{y}_{n\times1} - \begin{pmatrix}0\\0\\ \vdots\\1\\ \vdots\\0\end{pmatrix}_{n\times1} \quad (11.4)$$

← k elemento

↑ etiqueta del ejem p6

one hot vector = $y_{n\times1}$

→ $\frac{\partial L}{\partial W} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial W} = \frac{\partial L}{\partial z} x$

$$\frac{\partial L}{\partial W_{n\times m}} = \frac{\partial L}{\partial z_{n\times1}}\cdot (X_{m\times1})^T \quad (11.5)$$

$$\frac{\partial L}{\partial W_{n\times m}} = (\hat y_{n\times1} - y_{n\times1})\cdot(X_{m\times1})^T \quad (11.6)$$

= (y_hat - y)@x.T

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

$$\frac{\partial L}{\partial b_{n\times1}} = \frac{\partial L}{\partial z_{n\times1}}\cdot(1)_{1\times1} = \frac{\partial L}{\partial z_{n\times1}}$$

$$= \hat y_{n\times1} - y_{n\times1} \quad (11.7)$$

Después se puede actualizar W y b: (6.3) ↓ (11.8)

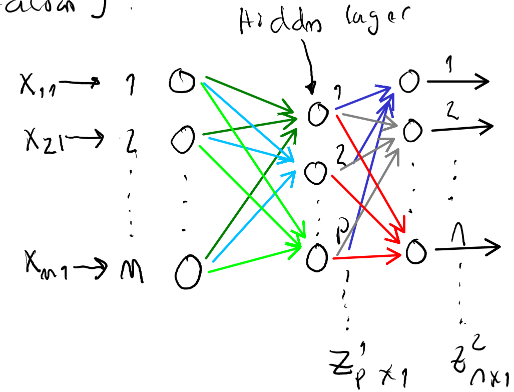
→ $W_{n\times m} = W_{n\times m} - \alpha \frac{\partial L}{\partial W_{n\times m}}$

↳ Ejm: 0.01

→ $b_{n\times1} = b_{n\times1} - \alpha\frac{\partial L}{\partial b_{n\times1}}$ (11.9)

↑ (6.4)

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



→ $z^1_{p\times1} = W^1_{p\times m} x_{m\times1} + b^1_{p\times1}$

→ $z^2_{n\times1} = W^2_{n\times p} z_{p\times1} + b^2_{n\times1}$

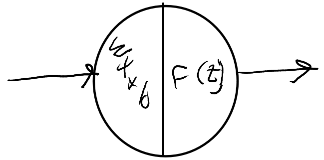
$= W^2_{n\times p}(W^1_{p\times m}x_{m\times1} + b^1_{p\times1}) + b^2_{n\times1}$

$= (W^2W^1)_{n\times m}X_{m\times1} + (W^2b^1)_{n\times1} + b^2_{n\times1}$

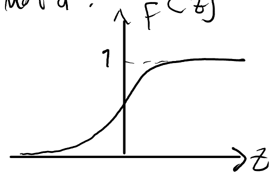
$= (W^2W^1)_{n\times m}x_{m\times1} + (W^2b^1+b^2)_{n\times1}$ (12.1)

Entonces las hidden layer no actuan, dando igual usar la sola capa de salida con n perceptrones

Funciones de activación:

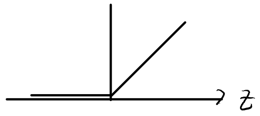


Sigmoid:



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

ReLU: Rectified Linear Unit:



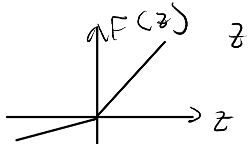
$z<0 \rightarrow f(z)=0$

$z\geq0 \rightarrow f(z)=z$

max(0, z) (12.3)

↑ máximo entre 0 y z

Leaky ReLU:



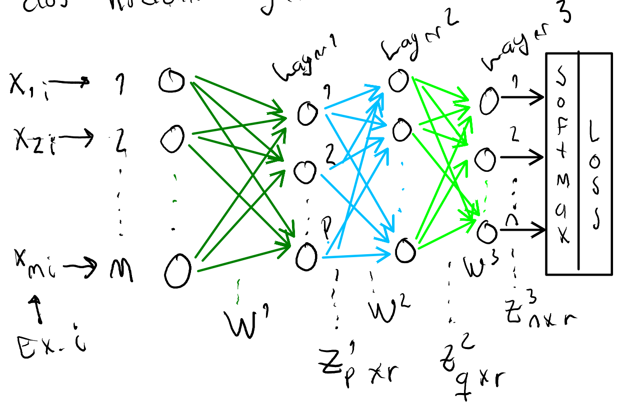
ejm ↓

$z<0 \rightarrow F(z) = -0.01z$

$z\geq0 \rightarrow F(z)=z$

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

Forwardpass

backpropagation

* Forwardpass:

$$z^1_{p\times r} = W^1_{p\times m} X_{m\times r} + \overbrace{(b^1_{p\times 1} \cdots b^1_{p\times 1})_{p\times r}}^{\text{Broadcast}}$$

Después de aplicar la función de activación de la capa 1:

$$a^1_{p\times r} = F^1(z^1_{p\times r})$$

→ $$z^2_{q\times r} = W^2_{q\times p} a^1_{p\times r} + \underbrace{(b^2_{q\times 1} \cdots b^2_{q\times 1})_{q\times r}}_{b^2_{q\times r}}$$

→ $$a^2_{q\times r} = F^2(z^2_{q\times r})$$

→ $$z^3_{n\times r} = W^3_{n\times q} a^2_{q\times r} + b^3_{n\times r}$$

(11.4)

Broadcast

$\Downarrow$

Softmax + Loss

etiqueta de los r ejemplos

* Backpropagation:

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

Softmax ← (4.2)

etiquetas

one hot vector para cada ejemplo:

$$\begin{array}{c c} & \begin{matrix} 1 & 2 & \cdots & r \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ \vdots \\ n \end{matrix} & \begin{pmatrix} 0 & 1 & & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & & 1 \end{pmatrix} \end{array}$$

→ $$\frac{\partial L}{\partial a^2} = \frac{\partial L}{\partial z^3}\frac{\partial z^3}{\partial a^2}$$

$$\frac{\partial L}{\partial a^2_{q\times r}} = (W^3_{n\times q})^T \frac{\partial L}{\partial z^3_{n\times r}}$$

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

$$\frac{\partial L}{\partial W^3_{n\times q}} = \frac{\partial L}{\partial z^3_{n\times r}}(a^2_{q\times r})^T$$

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

$$\frac{\partial L}{\partial b^3_{n\times r}} = \frac{\partial L}{\partial z^3_{n\times r}} \cdot 1$$

→ $$\frac{\partial L}{\partial z^2} = \frac{\partial L}{\partial a^2}\frac{\partial a^2}{\partial z^2}$$

$$\frac{\partial L}{\partial z^2_{q\times r}} = \frac{\partial L}{\partial a^2_{q\times r}}\left(\frac{\partial F^2(z^2)}{\partial z^2}\right)$$

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

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

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

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

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

$$\frac{\partial L}{\partial b^2_{q\times r}} = \frac{\partial L}{\partial z^2_{q\times r}} \cdot 1$$

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

$$\frac{\partial L}{\partial z^1_{p\times r}} = \frac{\partial L}{\partial a^1_{p\times r}}\left(\frac{\partial F^1(z^1)}{\partial z^1}\right)$$

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

$$\frac{\partial L}{\partial W^1_{p\times m}} = \frac{\partial L}{\partial z^1_{p\times r}}(X_{m\times r})^T$$

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

$$\frac{\partial L}{\partial b^1_{p\times r}} = \frac{\partial L}{\partial z^1_{p\times r}} \cdot 1$$

Usando Matriz Jacobiana:

$\partial L_r \rightarrow$ Loss función de r ejemplos

$$\rightarrow \frac{\partial L_r}{\partial z^3_n} = \begin{pmatrix} \frac{\partial L_1}{\partial z^3_1} & \cdots & \frac{\partial L_1}{\partial z^3_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial L_r}{\partial z^3_1} & \cdots & \frac{\partial L_r}{\partial z^3_n} \end{pmatrix} = \frac{\partial L}{\partial z_{r\times n}}$$

$$\frac{\partial L}{\partial z^3_{r\times n}} = \hat{y}_{r\times n} - \underset{\text{one hot vectors!}}{y_{r\times n}}$$

$$\begin{pmatrix} 0 & 1 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

$$\rightarrow \frac{\partial L_r}{\partial a^2_q} = \frac{\partial L_r}{\partial z^3_n}\frac{\partial z^3_n}{\partial a^2_q}$$

$$\frac{\partial L}{\partial a^2_{r\times q}} = \frac{\partial L}{\partial z^3_{r\times n}}\frac{\partial z^3}{\partial a^2_{n\times q}}$$

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

$$\rightarrow \frac{\partial L_r}{\partial W^3_{n\times q}} = \frac{\partial L_r}{\partial z^3_n}\frac{\partial z^3_n}{\partial W^3_{n\times q}}$$

$$\frac{\partial L}{\partial W^3_{r\times n\times q}} = (\hat{y}_{r\times n} - y_{r\times n})\, a^2_q$$

$\hookrightarrow$ Producto tensorial?

$$\rightarrow \frac{\partial L_r}{\partial b^3_n} = \frac{\partial L_r}{\partial z^3_n}\frac{\partial z^3_n}{\partial b^3_n}$$

$$\frac{\partial L}{\partial b^3_{r\times n}} = \frac{\partial L}{\partial z^3_{r\times n}} I_{n\times n} = \frac{\partial L}{\partial z^3_{r\times n}}$$

$$\rightarrow \frac{\partial L_r}{\partial W^2_{q\times p}} = \frac{\partial L_r}{\partial z^3_n}\frac{\partial z^3_n}{\partial a^2_q}\frac{\partial a^2_q}{\partial z^2_q}\frac{\partial z^2_q}{\partial W^2_{q\times p}}$$

$$= \underbrace{(\hat{y}_{r\times n} - y_{r\times n})\, W^3_{n\times q}\left(\frac{\partial F^2(z^2)}{\partial z^2}\right)_{q\times q}}_{r\times q} a^2_p$$

$$= \frac{\partial L}{\partial W^2_{r\times q\times p}}$$

$$\rightarrow \frac{\partial L_r}{\partial b^2_q} = \underbrace{\frac{\partial L_r}{\partial z^3_n}\frac{\partial z^3_n}{\partial a^2_q}\frac{\partial a^2_q}{\partial z^2_q}}_{\frac{\partial L_r}{\partial z^2_q}}\underbrace{\frac{\partial z^2_q}{\partial b^2_q}}_{I_{q\times q}}$$

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

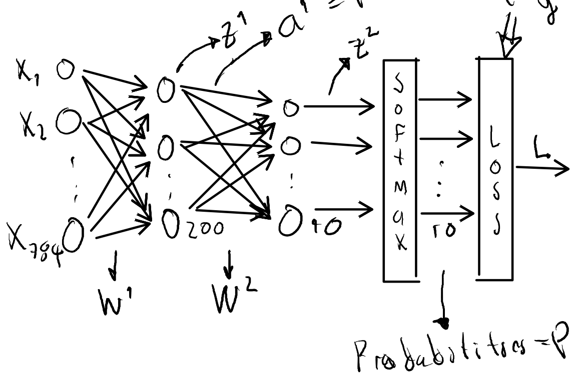
$$\rightarrow \frac{\partial L_r}{\partial W^1_{p\times m}} = \frac{\partial L_r}{\partial z^3_n}\frac{\partial z^3_n}{\partial a^2_q}\frac{\partial a^2_q}{\partial z^2_q}\frac{\partial z^2_q}{\partial a^1_p} \cdot \frac{\partial a^1_p}{\partial z^1_p}\frac{\partial z^1_p}{\partial W^1_{p\times m}}$$

$$\frac{\partial L}{\partial W^1_{r\times p\times m}} = \underbrace{\frac{\partial L_r}{\partial z^2_q} W^2_{q\times p}\left(\frac{\partial F^1(z^1)}{\partial z^1}\right)_{p\times p}}_{\frac{\partial L_r}{\partial z^1_p}} \cdot X_m$$

$$\rightarrow \frac{\partial L_r}{\partial b^1_p} = \frac{\partial L_r}{\partial z^1_p}\underbrace{\frac{\partial z^1_p}{\partial b^1_p}}_{I_{p\times p}}$$

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

Ex: MNIST:



→ $X_{64\times784}$ ← Pixels

Batch

→ $z^1_{200\times64}$

→ $a^1_{200\times64}$

→ $z^2_{10\times64}$

→ $P_{10\times64}$

→ $L_{1\times64}$ → cost ← escalar.

→ $W^1_{200\times784}$

→ $b^1_{200\times1}$

→ $W^2_{10\times200}$

→ $b^2_{10\times1}$

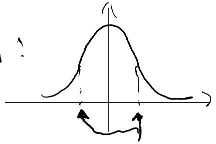
Training

Inicialmente:

$W^1_{200\times784} =$

np.random.randn(200,784)·0.001

Distribuição normal:



$\sigma = 0.001$

→ $b^1_{200\times1}$ = np.zeros((200,1))

→ $W^2_{10\times200} =$

np.random.randn(10,200)·0.001

→ $b^2_{10\times1}$ = np.zeros((10,1))

Forward:

$$z^1_{200\times64} = W^1_{200\times784}(X_{64\times784})^T + (\underset{1}{b^1}\; b^1 \dots \underset{64}{b^1})_{200\times64}$$

→ $a^1_{200\times64} = ReLU(z^1_{200\times64})$

→ $z^2_{10\times64} = W^2_{10\times200}\, a^1_{200\times64} + (b^2\; b^2 \dots b^2)_{10\times64}$

$$P_{10\times64} = \frac{(e^{z^2})_{10\times64}}{\underbrace{\left(\sum_j e^{z^2_{ji}}\right)_{1\times64}}_{\text{soma cada coluna de } (e^{z^2})_{10\times64} \text{ (ejm)}}} = \hat{y}_{10\times64} \quad (4.1)$$

(5.2)

$$(e^L)_{1\times64} = (P_{a,1}, P_{b,2} \dots P_{z,64})$$

etiqueta $e_{j,1}$ etiqueta $e_{j,2}$

→ $L_{1\times64} = Ln(e^L)_{1\times64}$

→ $cost = \frac{\sum_i^{64} -L_{1i}}{64}$ (5.3)

Backward:

$\hat{y}_{10\times64}$

→ $\frac{\partial L}{\partial z^2}_{10\times64} = P_{10\times64} - y_{10\times64}$

one hot vector para cada exemplo:

etiquetas

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

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

$$\frac{\partial L}{\partial W^2_{10\times200}} = \frac{\frac{\partial L}{\partial z^2_{10\times64}} (a^1_{200\times64})^T}{64}$$

→ $\frac{\partial L}{\partial b^2} = \frac{\partial L}{\partial z^2} \cancel{\frac{\partial z^2}{\partial b^2}}\; 1$

$$\frac{\partial L}{\partial b^2_{10\times1}} = \begin{pmatrix} \sum_j^{64} \frac{\partial L}{\partial z^2_{1j}} \\ \vdots \\ \sum_j^{64} \frac{\partial L}{\partial z^2_{10j}} \end{pmatrix}_{10\times1} / 64$$

soma elemen. da cada fila

$$\rightarrow \frac{\partial L}{\partial a^1} = \frac{\partial L}{\partial z^2} \frac{\partial z^2}{\partial a_1}$$

$$\frac{\partial L}{\partial a^1_{200 \times 64}} = \left(W^2_{10 \times 200}\right)^T \frac{\partial L}{\partial z^2_{10 \times 64}}$$

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

$$\frac{\partial L}{\partial z^1_{200 \times 64}} = \frac{\partial L}{\partial a^1_{200 \times 64}}$$