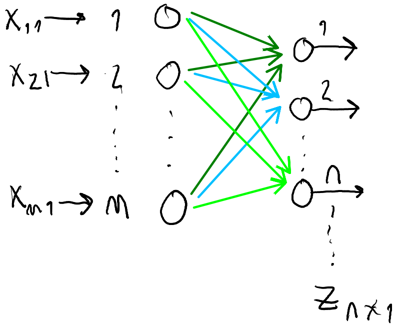


pesos * input + biases = $Z_{n\times 1}$

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

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

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

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

batch

p predictions a la vez

Función de activación softmax a la salida: → probabilidad

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

→ $k = 1, \ldots, n$

→ $j = 1, \ldots, n$

Usando batch: (4.2)

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

→ $k = 1, \ldots, n$ → $j = 1, \ldots, n$

→ $i = 1, \ldots, p$

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

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

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

Loss Function para el ejemplo i = Cross entropy = Xentropy

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

↓ valor correcto de la salida

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

$$y_{ki} = 0, 0, \ldots, 0, 1, 0, \ldots, \overset{n}{0}$$

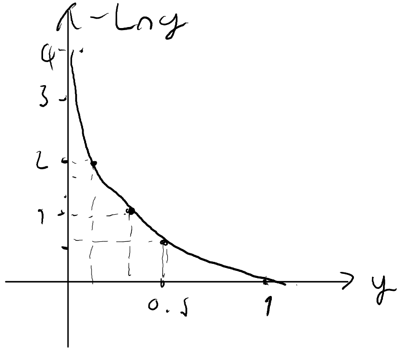
↑ k element ↓ clase esperada (etiqueta)

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

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

De la salida esperada para el ej. nº i

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



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:

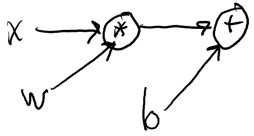
$$x \to \bigcirc \xrightarrow{w} \bigcirc \to z$$

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

$$\downarrow$$

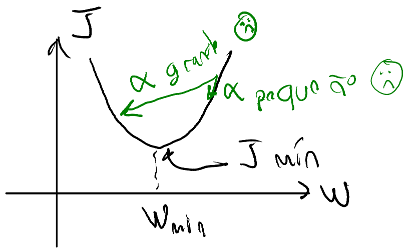
$$J(w, b)$$

Gráfica computacional: Cada operación es un nodo



Gradient descent:

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



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

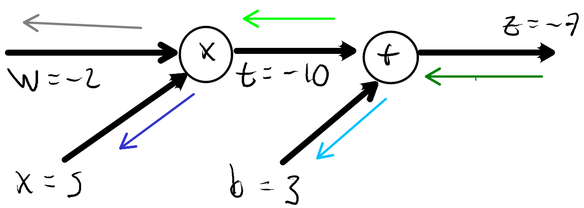
$\alpha \to$ learning rate = step size

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

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

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

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

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

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

$$z = -6.5 \checkmark$$

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:

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

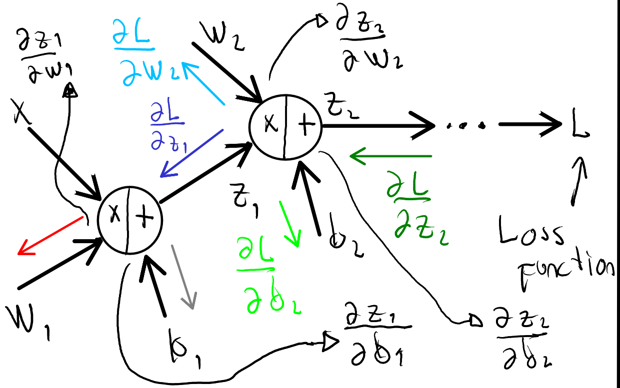
$\rightarrow z = t_0 + b = (-10) + (3.1) = -6.9$

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

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

Para dos perceptrons:

Gráfica computacional



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

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

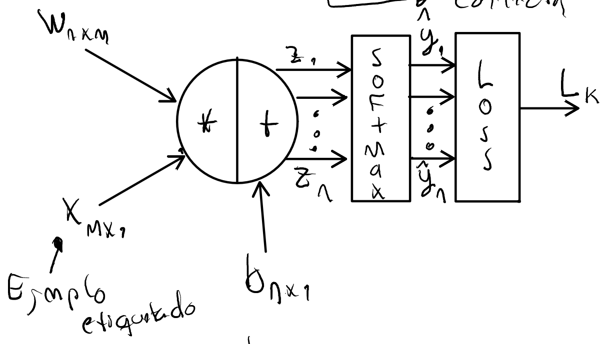
$$\rightarrow \frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial z_2}\frac{\partial z_2}{\partial w_2} \qquad \rightarrow w_2 = w_2 - \alpha \frac{\partial L}{\partial w_2}$$

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

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

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

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



$$W_{n \times m} X_{m \times 1} + b_{n \times 1} = z_{n \times 1}$$

↑ scores

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

$(0 \; 0 \ldots 1 \ldots 0)$

↑ 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:

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

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

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

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

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

← k elemento

etiqueta del ejemplo

one hot vector $= y_{n\times 1}$

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

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

$$\frac{\partial L}{\partial W_{n\times m}} = (\hat{y}_{n\times 1} - y_{n\times 1}) \cdot (X_{m\times 1})^T \quad (11.6)$$

= (y_hat-y)@ x.T

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

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

$$= \hat{y}_{n\times 1} - y_{n\times 1} \quad (11.7)$$

Después se puede actualizar W y b:

$$\rightarrow W_{n\times m} = W_{n\times m} - \alpha \frac{\partial L}{\partial W_{n\times m}} \quad \underset{(11.8)}{\overset{(6.3)}{\downarrow}}$$

α → Ejm: 0.01

$$\rightarrow b_{n\times 1} = b_{n\times 1} - \alpha \frac{\partial L}{\partial b_{n\times 1}} \quad \underset{(6.4)}{\overset{(11.9)}{\uparrow}}$$

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 esperada para el ejemplo i

Using Jacobian matriz:

$$\frac{\partial L_r}{\partial z_n} = \begin{pmatrix} \frac{\partial L_1}{\partial z_1^3} & \cdots & \frac{\partial L_1}{\partial z_n^3} \\ \vdots & \ddots & \vdots \\ \frac{\partial L_r}{\partial z_1^3} & \cdots & \frac{\partial L_r}{\partial z_n^3} \end{pmatrix} = \frac{\partial L}{\partial z_{r\times n}}$$

For row i (example i) of r examples:

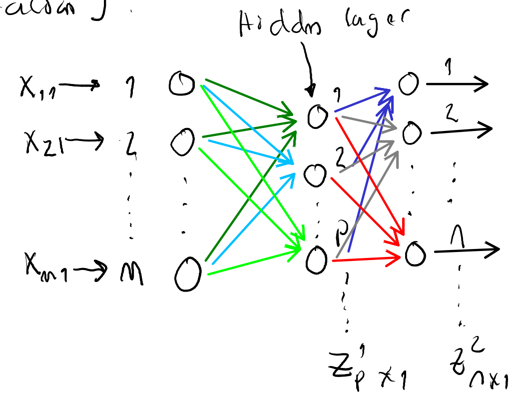
$$\frac{\partial L}{\partial z_{1\times n}} = \hat{y}_{1\times n} - y_{1\times n}$$

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

one hot vectors:

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

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



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

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

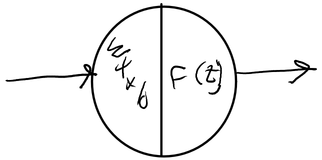
$= W^2_{n\times p}\left(W^1_{p\times m}\, x_{m\times1} + b^1_{p\times1}\right) + b^2_{n\times1}$

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

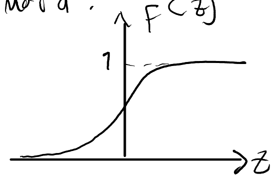
$= (W^2 W^1)_{n\times m}\, x_{m\times1} + (W^2 b^1 + b^2)_{n\times1}$ (12.1)

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

Funciones de activación:



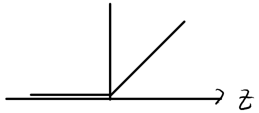
Sigmoid:



(12.2)

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

ReLU: Rectified Linear Unit:



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

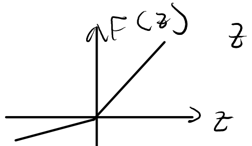
$z \geq 0 \rightarrow f(z) = z$

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

máximo entre 0 y z

Leaky ReLU:

ej

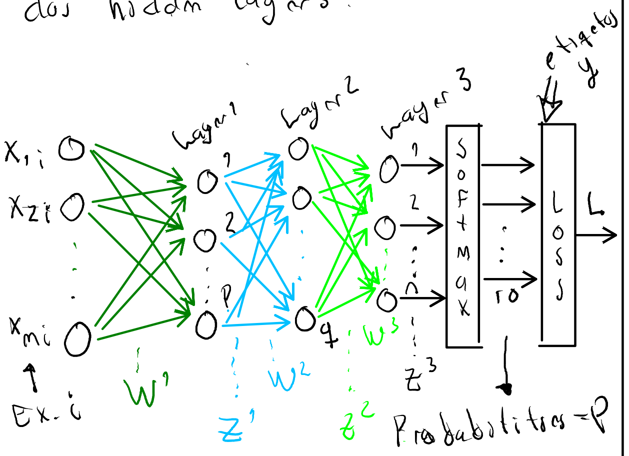


$z < 0 \rightarrow f(z) = -0.01z$

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

$\max(0.01z, z)$ (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

→ $X_{m\times r}$ ← Pixels
↑ Batch

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

Training

→ $z^1_{p\times r}$ → $a^1_{p\times r}$

→ $z^2_{q\times r}$ → $a^2_{q\times r}$

→ $z^3_{n\times r}$

→ $P_{n\times r}$

→ $L_{1\times r}$ → cost
escalar.

Forward pass:

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

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

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

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

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

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

(11.4) Broadcast

Softmax:

$$P_{n\times r} = \frac{(e^{z^3})_{n\times r}}{\left(\underbrace{\sum_j^n e^{z^3_{ji}}}_{j}\right)_{1\times r}} = \hat{y}_{n\times r} \quad (4.9)$$

Suma cada columna de $(e^{z^3})_{n\times r}$ (ejm)

Cross entropy:

$$\to L_{1\times r} = -\ln(P_{a,1}, P_{b,2} \cdots P_{z,r})_{1\times r}$$

etiqueta ej. 1 — etiqueta ej. 2

$$\to \text{cost} = \frac{\sum_i^r L_{1i}}{r} \quad (5.3)$$

Backpropagation:

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

$\hat{y}_{n\times r}$

etiquetas

one hot vector por cada ejemplo:

$$\begin{array}{c|c} & \\ \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 \cdots & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 1 \end{pmatrix} & \begin{array}{l} \leftarrow 1 \\ \leftarrow 2 \\ \vdots \\ \leftarrow n \end{array} \\ \begin{array}{ccc} 1 & 2 \cdots & r \end{array} & \end{array}$$

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

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

must be normalized

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

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

normalized

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

sum elements of each row

$$\frac{\partial L}{\partial b^3_{n\times 1}} = \begin{pmatrix} \sum_j^r \frac{\partial L}{\partial z^3_{1j}} \\ \vdots \\ \sum_j^r \frac{\partial L}{\partial z^3_{nj}} \end{pmatrix}_{n\times 1} \Big/ r$$

normalized

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

W^3 normalized

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

$$= \frac{\partial f^2(z^2)}{\partial z^2} = g(z^2)$$

$$g(z^2_{ij})$$

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

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

must be normalized

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

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

$$\frac{\partial L}{\partial z^2} \leftarrow \frac{\partial L}{\partial a^2}$$

W^3 must be normalized

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

$$\frac{\partial L}{\partial b^2_{q\times 1}} = \begin{pmatrix} \sum_j^r \frac{\partial L}{\partial z^2_{1j}} \\ \vdots \\ \sum_j^r \frac{\partial L}{\partial z^2_{qj}} \end{pmatrix}_{q\times 1} \Big/ \cancel{r}$$

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

W^2 normalized

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

W^2 normalized

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

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

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

$$\frac{\partial L}{\partial b^1_{p\times 1}} = \begin{pmatrix} \sum_j^r \frac{\partial L}{\partial z^1_{1j}} \\ \vdots \\ \sum_j^r \frac{\partial L}{\partial z^1_{pj}} \end{pmatrix}_{p\times 1}$$

Training for each epoch, for each batch:

$$\rightarrow W^1_{p\times m} = W^1_{p\times m} - \alpha \frac{dL}{dW^1_{p\times m}}$$

$$\rightarrow b^1_{p\times 1} = b^1_{p\times 1} - \alpha \frac{dL}{db^1_{p\times 1}}$$

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

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

$\rightarrow W^3_{n \times q} = W^3_{n \times q} - \alpha \frac{dL}{dW^3_{n \times q}}$

$\rightarrow b^3_{n \times 1} = b^3_{n \times 1} - \alpha \frac{dL}{db^3_{n \times 1}}$

Accuracy using validation data:
after each epoch

Forward (Validation Data)

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

$\downarrow$

$P_{n \times r}$:

Predictions $= Pe = (P_{max,1} \cdots P_{max,r})_{1 \times r}$

max row of col 1.   max row col. r

$$\text{Accuracy} = \frac{\sum_{i=1}^{\#\text{batches}} \sum_{j=1}^{r} \delta_{Pe_j = \text{Label}_j}}{\#\text{ Validation data}}$$

Validation data

$\hookrightarrow$ %

$\hookrightarrow = \#\text{batches} \times r$

Prediction of a image after

\# epochs: $\rightarrow$

Optimal $\left\{ \boxed{\begin{array}{ll} \rightarrow W^1_{p \times m} & \rightarrow b^1_{p \times 1} \\ \rightarrow W^2_{q \times p} & \rightarrow b^2_{q \times 1} \\ \rightarrow W^3_{n \times q} & \rightarrow b^3_{n \times 1} \end{array}} \right.$

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

$\hookrightarrow$ Total pixels

$\downarrow$

Forward $(X_{1 \times m})$

$\downarrow$

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

$\downarrow$ softmax

$\hat{y}_{n \times 1} = P_{n \times 1}$

$\downarrow$

Prediction $= Pe = P_{max}$ of $P_{n \times 1}$

Usando Matriz Jacobiana:

$\frac{\partial L_r}{\partial z^3_n}$ → Loss funtion de r ejemplos

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

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

one hot vectors!

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

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

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

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

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

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

↳ Producto tensorial?

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

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

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

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

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

$r\times q$ ← $\frac{\partial L_r}{\partial z^2_q}$

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

↳ $I_{q\times q}$

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

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

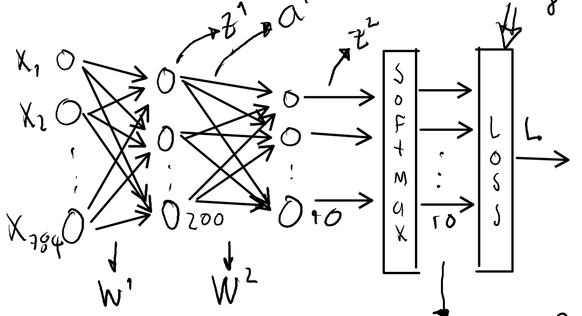
$$\frac{\partial L}{\partial W^1_{r\times p\times m}} = \underbrace{\frac{\partial L_r}{\partial z^2_q} W^2_{q\times p} \left(\frac{\partial f^1(z^1)}{\partial z^1}\right)_{p\times p}}_{\frac{\partial L_r}{\partial z^1_p}} \cdot X_m$$

→ $$\frac{\partial L_r}{\partial b^1_p} = \frac{\partial L_r}{\partial z^1_p} \frac{\partial z^1_p}{\partial b^1_p}$$

↳ $I_{p\times p}$

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

Ex: MNIST:



Probabilities = P

→ $X_{64\times784}$ ← Pixels
↑ Batch

→ $z^1_{200\times64}$

→ $a^1_{200\times64}$

→ $z^2_{10\times64}$

→ $P_{10\times64}$

→ $L_{1\times64}$ → cost ← escalar.

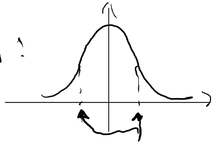
→ $W^1_{200\times784}$
→ $b^1_{200\times1}$
→ $W^2_{10\times200}$
→ $b^2_{10\times1}$

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:

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

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

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

Softmax:

$$P_{10\times64} = \frac{(e^{z^2})_{10\times64}}{\Big(\underbrace{\sum_j^{10} e^{z^2_{ji}}}\Big)_{1\times64}} = \hat{y}_{10\times64} \quad (4.1)$$

Soma cada coluna de $(e^{z^2})_{10\times64}$ (ej,m)

Cross entropy:

(5.2)
$$(e^L)_{1\times64} = (P_{a1}, P_{b2} \ldots P_{z64})$$
etiqueta ej. 1 etiqueta ej. 2

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

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

Backward:

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

etiquetas

one hot vector por cada exemplo:

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

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

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

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

→ sum elements of each row

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

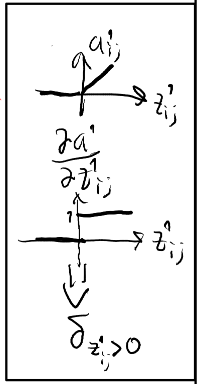
$$\rightarrow \frac{\partial L}{\partial a^1} = \frac{\partial L}{\partial z^2}\frac{\partial z^2}{\partial a^1}$$

must be normalized

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

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

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 of $\left(\frac{\partial L}{\partial a^1}\right)_{200\times 64}$

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

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

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

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

must be normalized

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

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

$$\frac{\partial L}{\partial b^1_{200\times 1}} = \begin{pmatrix} \sum_j^{64} \frac{\partial L}{\partial z^1_{1j}} \\ \vdots \\ \sum_j^{64} \frac{\partial L}{\partial z^1_{200j}} \end{pmatrix}_{200\times 1} / \cancel{64}$$

Training for each epoch, for each batch:

$\rightarrow \alpha = 0.001$

$$\rightarrow W^1_{200\times 784} = W^1_{200\times 784} - \alpha \frac{dL}{dW^1_{200\times 784}}$$

$$\rightarrow b^1_{200\times 1} = b^1_{200\times 1} - \alpha \frac{dL}{db^1_{200\times 1}}$$

$$\rightarrow W^2_{10\times 200} = W^2_{10\times 200} - \alpha \frac{dL}{dW^2_{10\times 200}}$$

$$\rightarrow b^2_{10\times 1} = b^2_{10\times 1} - \alpha \frac{dL}{db^2_{10\times 1}}$$

Accuracy using validation data: after each epoch

Forward (Validation Data)

$\downarrow$

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

$\downarrow$

$P_{10\times 64}$:

Predictions $= Pe = (P_{max,1} \cdots P_{max,64})_{1\times 64}$

max row of col 1, max row col. 64

$$\text{Accuracy} = \frac{\sum_{i=1}^{\#\text{batches}} \sum_{j=1}^{64} \delta_{Pe_j = \text{Label}_j}}{\#\text{ Validation data}}$$

Validation data

Prediction of a image after # epochs:

$$\text{Optimal} \begin{cases} \rightarrow W^1_{200\times 784} \\ \rightarrow b^1_{200\times 1} \\ \rightarrow W^2_{10\times 200} \\ \rightarrow b^2_{10\times 1} \end{cases}$$

$\rightarrow \text{image}_{1\times 784} = X_{1\times 784}$

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

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

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

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

Prediction $= Pe = P_{max}$ of $P_{10\times 1}$