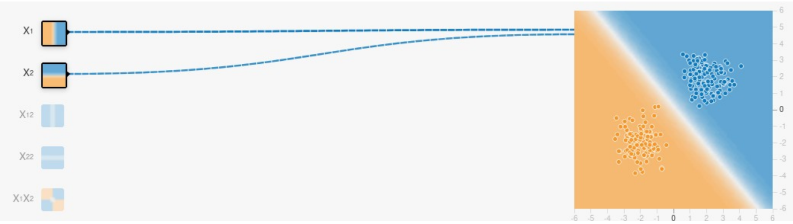
4.2 Red Neuronal: aprende características

- Con un modelo lineal + la sigmoide:

$$y=\sigma(\phi_1 x_1+\phi_2 x_2+\phi_0)=\frac{1}{1+e^{-(\phi_1 x_1+\phi_2 x_2+\phi_0)}}$$

Puede ser suficiente:



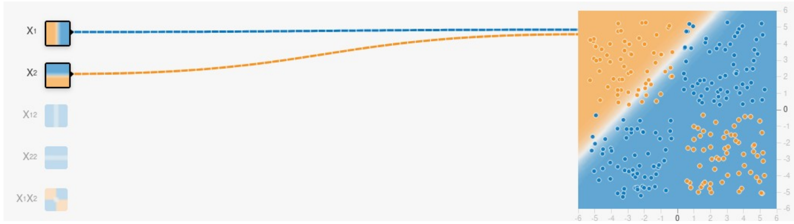
https://playground.tensorflow.org

4.2 Red Neuronal: aprende características

- Con un modelo lineal + la sigmoide:

$$y=\sigma(\phi_1 x_1+\phi_2 x_2+\phi_0)=\frac{1}{1+e^{-(\phi_1 x_1+\phi_2 x_2+\phi_0)}}$$

O puede ser deficiente:

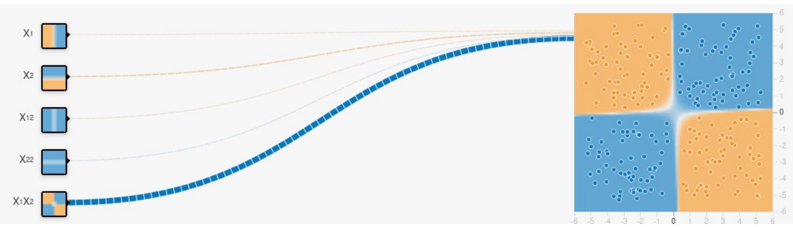


https://playground.tensorflow.org

4.2 Red Neuronal: aprende características

- Con las características adecuadas al problema :
  - p.ejemplo: (x_1, x_2, x_1^2, x_2^2, $x_1 \cdot x_2$)
- Un modelo lineal en los parámetros (ahora no lineal en x_1 y x_2) funcionará siempre:

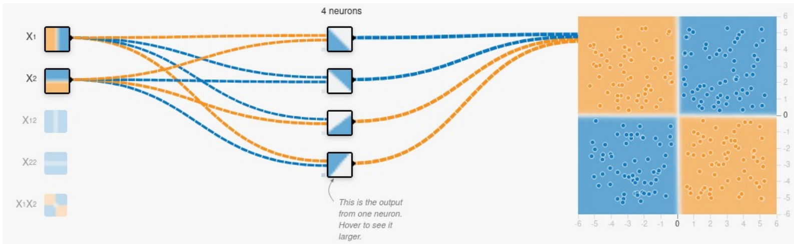
$$y=\sigma(\phi_5 x_1\cdot x_2+\phi_4 x_2^2+\phi_3 x_1^2+\phi_1 x_1+\phi_2 x_2+\phi_0)=\frac{1}{1+e^{-(\phi_5 x_1\cdot x_2+\phi_4 x_2^2+\phi_3 x_1^2+\phi_1 x_1+\phi_2 x_2+\phi_0)}}$$



https://playground.tensorflow.org

4.2 Red Neuronal: aprende características

- Las neuronas de la capa oculta funcionan como “características aprendidas” para el problema:



https://playground.tensorflow.org

4.3 Backpropagation en Redes Neuronales

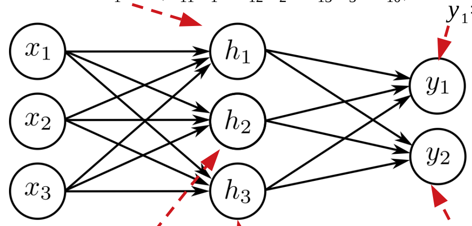
- Grafos de cómputo en Redes Neuronales
- Retropropagación en Redes Neuronales
- Eficiencia con el algoritmo de retropropagación

Grafos de cómputo en Redes de Neuronas

- El grafo de la red de neuronas es un grafo de cómputo:

$h_1 = a(\theta_{11} x_1 + \theta_{12} x_2 + \theta_{13} x_3 + \theta_{10})$

$y_1 = \phi_{11} h_1 + \phi_{12} h_2 + \phi_{13} h_3 + \phi_{10}$



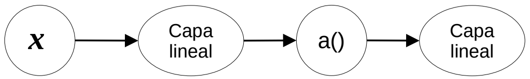
$y_2 = \phi_{21} h_1 + \phi_{22} h_2 + \phi_{23} h_3 + \phi_{20}$

$h_3 = a(\theta_{31} x_1 + \theta_{32} x_2 + \theta_{33} x_3 + \theta_{30})$

$h_2 = a(\theta_{21} x_1 + \theta_{22} x_2 + \theta_{23} x_3 + \theta_{20})$

Es un grafo con demasiado detalle y se suele expresar de manera más compacta

Grafos de cómputo en Redes de Neuronas

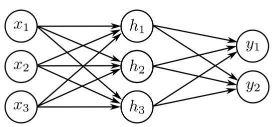


3x1

$\boldsymbol{z} = \boldsymbol{\Omega}_0 \boldsymbol{x} + \beta_0$ (3x1, 3x3, 3x1)

$\boldsymbol{h} = a(\boldsymbol{z})$ (3x1)

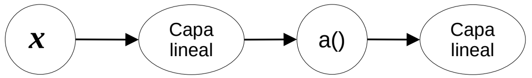
$\boldsymbol{y} = \boldsymbol{\Omega}_1 \boldsymbol{h} + \beta_1$ (2x3, 2x1)



4.3 Backpropagation en Redes Neuronales

- Grafos de cómputo en Redes Neuronales
- Retropropagación en Redes Neuronales
- Eficiencia con el algoritmo de retropropagación

Grafos de cómputo en Redes de Neuronas

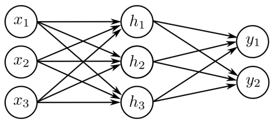


3x1

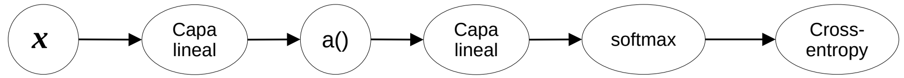
$z=\Omega_0 x+\beta_0$ (3x1, 3x3, 3x1)

$h=a(z)$ (3x1)

$y=\Omega_1 h+\beta_1$ (2x3, 2x1)



Retropropagación en Redes Neuronales



3x1

$z=\Omega_0 x+\beta_0$ (3x1, 3x3, 3x1)

$h=a(z)$ (3x1)

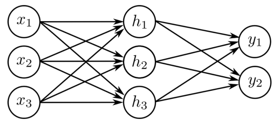
$y=\Omega_1 h+\beta_1$ (2x3, 2x1)

s (2x1)

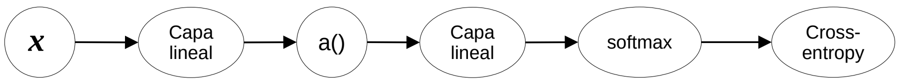
J (1x1)

$$s(y)=\begin{bmatrix}\frac{e^{y_1}}{e^{y_1}+e^{y_2}}\\ \frac{e^{y_2}}{e^{y_1}+e^{y_2}}\end{bmatrix}$$

$$J(l,s)=-\sum_{i=1}^{2} I(l_i=i)\log(s_i)$$



Retropropagación en Redes Neuronales



3x1

$z=\Omega_0 x+\beta_0$ (3x1, 3x3, 3x1)

$h=a(z)$ (3x1)

$y=\Omega_1 h+\beta_1$ (2x3, 2x1)

s (2x1)

J (1x1)

$$s(y)=\begin{bmatrix}\frac{e^{y_1}}{e^{y_1}+e^{y_2}}\\ \frac{e^{y_2}}{e^{y_1}+e^{y_2}}\end{bmatrix}$$

$$J(l,s)=-\sum_{i=1}^{2} I(l_i=i)\log(s_i)$$

$$\frac{\partial z(x)}{\partial x}=\Omega_0^T$$

$$\frac{\partial y(h)}{\partial h}=\Omega_1^T$$

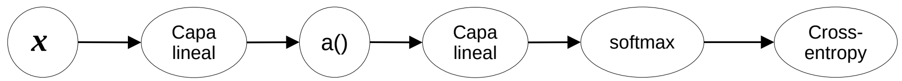
$$h=\begin{bmatrix}ReLU(z_1)\\ ReLU(z_2)\\ ReLU(z_3)\end{bmatrix}=\begin{bmatrix}máx(0,z_1)\\ máx(0,z_2)\\ máx(0,z_3)\end{bmatrix}$$

$$\frac{\partial J(l,s)}{\partial s}=\begin{bmatrix}-I(l_i=1)\cdot\frac{1}{s_1}\\ -I(l_i=2)\cdot\frac{1}{s_2}\end{bmatrix}$$

$$\frac{\partial a(z)}{\partial z}=\begin{bmatrix}I(z_1>0) & 0 & 0\\ 0 & I(z_2>0) & 0\\ 0 & 0 & I(z_3>0)\end{bmatrix}$$

$$\frac{\partial s(y)}{\partial y}=\begin{bmatrix}s_1\cdot(1-s_1) & -s_1\cdot s_2\\ -s_1\cdot s_2 & s_2\cdot(1-s_2)\end{bmatrix}$$

Retropropagación en Redes Neuronales



3x1

$z=\Omega_0 x+\beta_0$ (3x1, 3x3, 3x1)

$h=a(z)$ (3x1)

$y=\Omega_1 h+\beta_1$ (2x3, 2x1)

s (2x1)

J (1x1)

$$s(y)=\begin{bmatrix}\frac{e^{y_1}}{e^{y_1}+e^{y_2}}\\ \frac{e^{y_2}}{e^{y_1}+e^{y_2}}\end{bmatrix}$$

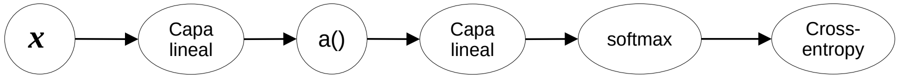
$$J(l,s)=-\sum_{i=1}^{2} I(l_i=i)\log(s_i)$$

Backpropagation:

$$\frac{\partial J}{\partial s}=\begin{bmatrix}-I(l_i=1)\cdot\frac{1}{s_1}\\ -I(l_i=2)\cdot\frac{1}{s_2}\end{bmatrix}$$

$$\boxed{\frac{\partial z(x)}{\partial x}=\Omega_0^T \quad \frac{\partial a(z)}{\partial z}=\begin{bmatrix}I(z_1>0) & 0 & 0\\ 0 & I(z_2>0) & 0\\ 0 & 0 & I(z_3>0)\end{bmatrix} \quad \frac{\partial y(h)}{\partial h}=\Omega_1^T \quad \frac{\partial s(y)}{\partial y}=\begin{bmatrix}s_1\cdot(1-s_1) & -s_1\cdot s_2\\ -s_1\cdot s_2 & s_2\cdot(1-s_2)\end{bmatrix} \quad \frac{\partial J(l,s)}{\partial s}=\begin{bmatrix}-I(l_i=1)\cdot\frac{1}{s_1}\\ -I(l_i=2)\cdot\frac{1}{s_2}\end{bmatrix}}$$

Retropropagación en Redes Neuronales



3x1 $\quad z=\Omega_0 x+\beta_0 \quad h=a(z) \quad y=\Omega_1 h+\beta_1 \quad s \quad J$

3x1 3x3 3x1 $\quad$ 3x1 $\quad$ 2x3 2x1 $\quad$ 2x1 $\quad$ 1x1

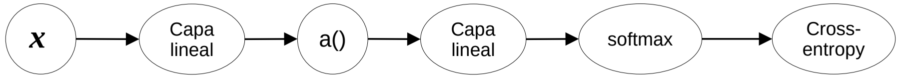
$$s(y)=\begin{bmatrix}\frac{e^{y_1}}{e^{y_1}+e^{y_2}}\\ \frac{e^{y_2}}{e^{y_1}+e^{y_2}}\end{bmatrix} \qquad J(l,s)=-\sum_{i=1}^{2} I(l_i=i)\log(s_i)$$

Backpropagation:

$$\frac{\partial J}{\partial y}=\begin{bmatrix}s_1\cdot(1-s_1) & -s_1\cdot s_2\\ -s_1\cdot s_2 & s_2\cdot(1-s_2)\end{bmatrix}\cdot\underbrace{\begin{bmatrix}-I(l_i=1)\cdot\frac{1}{s_1}\\ -I(l_i=2)\cdot\frac{1}{s_2}\end{bmatrix}}_{\frac{\partial J}{\partial s}}=\frac{\partial s}{\partial y}\cdot\frac{\partial J}{\partial s}$$

$$\boxed{\frac{\partial z(x)}{\partial x}=\Omega_0^T \quad \frac{\partial a(z)}{\partial z}=\begin{bmatrix}I(z_1>0)&0&0\\0&I(z_2>0)&0\\0&0&I(z_3>0)\end{bmatrix} \quad \frac{\partial y(h)}{\partial h}=\Omega_1^T \quad \frac{\partial s(y)}{\partial y}=\begin{bmatrix}s_1\cdot(1-s_1) & -s_1\cdot s_2\\ -s_1\cdot s_2 & s_2\cdot(1-s_2)\end{bmatrix} \quad \frac{\partial J(l,s)}{\partial s}=\begin{bmatrix}-I(l_i=1)\cdot\frac{1}{s_1}\\ -I(l_i=2)\cdot\frac{1}{s_2}\end{bmatrix}}$$

Retropropagación en Redes Neuronales



3x1 $\quad z=\Omega_0 x+\beta_0 \quad h=a(z) \quad y=\Omega_1 h+\beta_1 \quad s \quad J$

3x1 3x3 3x1 $\quad$ 3x1 $\quad$ 2x3 2x1 $\quad$ 2x1 $\quad$ 1x1

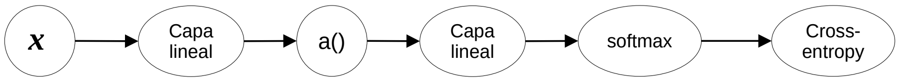
$$s(y)=\begin{bmatrix}\frac{e^{y_1}}{e^{y_1}+e^{y_2}}\\ \frac{e^{y_2}}{e^{y_1}+e^{y_2}}\end{bmatrix} \qquad J(l,s)=-\sum_{i=1}^{2} I(l_i=i)\log(s_i)$$

Backpropagation:

$$\frac{\partial J}{\partial h}=\Omega_1^T\cdot\underbrace{\begin{bmatrix}s_1\cdot(1-s_1) & -s_1\cdot s_2\\ -s_1\cdot s_2 & s_2\cdot(1-s_2)\end{bmatrix}\cdot\begin{bmatrix}-I(l_i=1)\cdot\frac{1}{s_1}\\ -I(l_i=2)\cdot\frac{1}{s_2}\end{bmatrix}}_{\frac{\partial J}{\partial y}}=\frac{\partial y}{\partial h}\cdot\frac{\partial J}{\partial y}$$

$$\boxed{\frac{\partial z(x)}{\partial x}=\Omega_0^T \quad \frac{\partial a(z)}{\partial z}=\begin{bmatrix}I(z_1>0)&0&0\\0&I(z_2>0)&0\\0&0&I(z_3>0)\end{bmatrix} \quad \frac{\partial y(h)}{\partial h}=\Omega_1^T \quad \frac{\partial s(y)}{\partial y}=\begin{bmatrix}s_1\cdot(1-s_1) & -s_1\cdot s_2\\ -s_1\cdot s_2 & s_2\cdot(1-s_2)\end{bmatrix} \quad \frac{\partial J(l,s)}{\partial s}=\begin{bmatrix}-I(l_i=1)\cdot\frac{1}{s_1}\\ -I(l_i=2)\cdot\frac{1}{s_2}\end{bmatrix}}$$

Retropropagación en Redes Neuronales



3x1 $\quad z=\Omega_0 x+\beta_0 \quad h=a(z) \quad y=\Omega_1 h+\beta_1 \quad s \quad J$

3x1 3x3 3x1 $\quad$ 3x1 $\quad$ 2x3 2x1 $\quad$ 2x1 $\quad$ 1x1

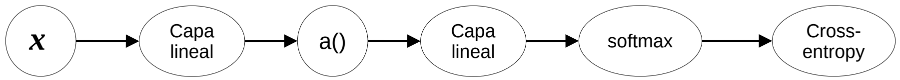
$$s(y)=\begin{bmatrix}\frac{e^{y_1}}{e^{y_1}+e^{y_2}}\\ \frac{e^{y_2}}{e^{y_1}+e^{y_2}}\end{bmatrix} \qquad J(l,s)=-\sum_{i=1}^{2} I(l_i=i)\log(s_i)$$

Backpropagation:

$$\frac{\partial J}{\partial z}=\begin{bmatrix}I(z_1>0)&0&0\\0&I(z_2>0)&0\\0&0&I(z_3>0)\end{bmatrix}\cdot\underbrace{\Omega_1^T\cdot\begin{bmatrix}s_1\cdot(1-s_1) & -s_1\cdot s_2\\ -s_1\cdot s_2 & s_2\cdot(1-s_2)\end{bmatrix}\cdot\begin{bmatrix}-I(l_i=1)\cdot\frac{1}{s_1}\\ -I(l_i=2)\cdot\frac{1}{s_2}\end{bmatrix}}_{\frac{\partial J}{\partial h}}=\frac{\partial h}{\partial z}\frac{\partial J}{\partial h}$$

$$\boxed{\frac{\partial z(x)}{\partial x}=\Omega_0^T \quad \frac{\partial a(z)}{\partial z}=\begin{bmatrix}I(z_1>0)&0&0\\0&I(z_2>0)&0\\0&0&I(z_3>0)\end{bmatrix} \quad \frac{\partial y(h)}{\partial h}=\Omega_1^T \quad \frac{\partial s(y)}{\partial y}=\begin{bmatrix}s_1\cdot(1-s_1) & -s_1\cdot s_2\\ -s_1\cdot s_2 & s_2\cdot(1-s_2)\end{bmatrix} \quad \frac{\partial J(l,s)}{\partial s}=\begin{bmatrix}-I(l_i=1)\cdot\frac{1}{s_1}\\ -I(l_i=2)\cdot\frac{1}{s_2}\end{bmatrix}}$$

Retropropagación en Redes Neuronales



3x1 $\quad z=\Omega_0 x+\beta_0 \quad h=a(z) \quad y=\Omega_1 h+\beta_1 \quad s \quad J$

3x1 3x3 3x1 $\quad$ 3x1 $\quad$ 2x3 2x1 $\quad$ 2x1 $\quad$ 1x1

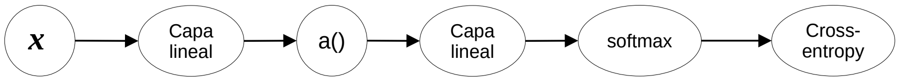
$$s(y)=\begin{bmatrix}\frac{e^{y_1}}{e^{y_1}+e^{y_2}}\\ \frac{e^{y_2}}{e^{y_1}+e^{y_2}}\end{bmatrix} \qquad J(l,s)=-\sum_{i=1}^{2} I(l_i=i)\log(s_i)$$

Backpropagation:

$$\frac{\partial J}{\partial x}=\Omega_0^T\cdot\underbrace{\begin{bmatrix}I(z_1>0)&0&0\\0&I(z_2>0)&0\\0&0&I(z_3>0)\end{bmatrix}\cdot\Omega_1^T\cdot\begin{bmatrix}s_1\cdot(1-s_1) & -s_1\cdot s_2\\ -s_1\cdot s_2 & s_2\cdot(1-s_2)\end{bmatrix}\cdot\begin{bmatrix}-I(l_i=1)\cdot\frac{1}{s_1}\\ -I(l_i=2)\cdot\frac{1}{s_2}\end{bmatrix}}_{\frac{\partial J}{\partial z}}=\frac{\partial z}{\partial x}\frac{\partial J}{\partial z}$$

$$\boxed{\frac{\partial z(x)}{\partial x}=\Omega_0^T \quad \frac{\partial a(z)}{\partial z}=\begin{bmatrix}I(z_1>0)&0&0\\0&I(z_2>0)&0\\0&0&I(z_3>0)\end{bmatrix} \quad \frac{\partial y(h)}{\partial h}=\Omega_1^T \quad \frac{\partial s(y)}{\partial y}=\begin{bmatrix}s_1\cdot(1-s_1) & -s_1\cdot s_2\\ -s_1\cdot s_2 & s_2\cdot(1-s_2)\end{bmatrix} \quad \frac{\partial J(l,s)}{\partial s}=\begin{bmatrix}-I(l_i=1)\cdot\frac{1}{s_1}\\ -I(l_i=2)\cdot\frac{1}{s_2}\end{bmatrix}}$$

Pregunta 4



$\mathbf{x}$ 3x1 — $\mathbf{z}=\boldsymbol{\Omega}_0\mathbf{x}+\beta_0$ (3x1, 3x3, 3x1) — $\mathbf{h}=a(\mathbf{z})$ 3x1 — $\mathbf{y}=\boldsymbol{\Omega}_1\mathbf{h}+\beta_1$ (2x3, 2x1) — $\mathbf{s}$ 2x1 — J 1x1

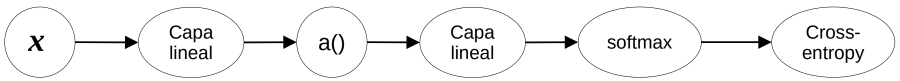
$$s(\mathbf{y})=\begin{bmatrix}\frac{e^{y_1}}{e^{y_1}+e^{y_2}}\\ \frac{e^{y_2}}{e^{y_1}+e^{y_2}}\end{bmatrix} \qquad J(\mathbf{l},\mathbf{s})=-\sum_{i=1}^{2} I(l_i=i)\log(s_i)$$

- ¿Falta algo por calcular?
- ¿Cuál es el objetivo último del algoritmo backpropagation en SGD?

4.3 Backpropagation en Redes Neuronales

- Grafos de cómputo en Redes Neuronales
- Retropropagación en Redes Neuronales
- Eficiencia con el algoritmo de retropropagación

Retropropagación en Redes Neuronales

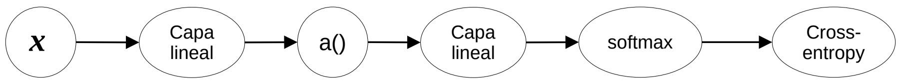


$\mathbf{x}$ 3x1 — $\mathbf{z}=\boldsymbol{\Omega}_0\mathbf{x}+\beta_0$ (3x1, 3x3, 3x1) — $\mathbf{h}=a(\mathbf{z})$ 3x1 — $\mathbf{y}=\boldsymbol{\Omega}_1\mathbf{h}+\beta_1$ (2x3, 2x1) — $\mathbf{s}$ 2x1 — J 1x1

$$\frac{\partial J}{\partial \mathbf{x}}=\underbrace{\frac{\partial z}{\partial \mathbf{x}}}_{3x3}\cdot\underbrace{\frac{\partial h}{\partial \mathbf{z}}}_{3x3}\cdot\underbrace{\frac{\partial y}{\partial \mathbf{h}}}_{3x2}\cdot\underbrace{\frac{\partial s}{\partial \mathbf{y}}}_{2x2}\cdot\underbrace{\frac{\partial J}{\partial \mathbf{s}}}_{2x1}$$

$$s(\mathbf{y})=\begin{bmatrix}\frac{e^{y_1}}{e^{y_1}+e^{y_2}}\\ \frac{e^{y_2}}{e^{y_1}+e^{y_2}}\end{bmatrix} \qquad J(\mathbf{l},\mathbf{s})=-\sum_{i=1}^{2} I(l_i=i)\log(s_i)$$

Retropropagación en Redes Neuronales



$\mathbf{x}$ 3x1 — $\mathbf{z}=\boldsymbol{\Omega}_0\mathbf{x}+\beta_0$ (3x1, 3x3, 3x1) — $\mathbf{h}=a(\mathbf{z})$ 3x1 — $\mathbf{y}=\boldsymbol{\Omega}_1\mathbf{h}+\beta_1$ (2x3, 2x1) — $\mathbf{s}$ 2x1 — J 1x1

$$\frac{\partial J}{\partial \mathbf{x}}=\underbrace{\frac{\partial z}{\partial \mathbf{x}}}_{3x3}\cdot\underbrace{\frac{\partial h}{\partial \mathbf{z}}}_{3x3}\cdot\underbrace{\frac{\partial y}{\partial \mathbf{h}}}_{3x2}\cdot\underbrace{\frac{\partial s}{\partial \mathbf{y}}}_{2x2}\cdot\underbrace{\frac{\partial J}{\partial \mathbf{s}}}_{2x1}$$

$$s(\mathbf{y})=\begin{bmatrix}\frac{e^{y_1}}{e^{y_1}+e^{y_2}}\\ \frac{e^{y_2}}{e^{y_1}+e^{y_2}}\end{bmatrix} \qquad J(\mathbf{l},\mathbf{s})=-\sum_{i=1}^{2} I(l_i=i)\log(s_i)$$

$$\boxed{\frac{\partial z(\mathbf{x})}{\partial \mathbf{x}}=\boldsymbol{\Omega}_0^T \quad \frac{\partial a(\mathbf{z})}{\partial \mathbf{z}}=\begin{bmatrix}I(z_1>0)&0&0\\0&I(z_2>0)&0\\0&0&I(z_3>0)\end{bmatrix} \quad \frac{\partial y(\mathbf{h})}{\partial \mathbf{h}}=\boldsymbol{\Omega}_1^T \quad \frac{\partial s(\mathbf{y})}{\partial \mathbf{y}}=\begin{bmatrix}s_1\cdot(1-s_1)&-s_1\cdot s_2\\-s_1\cdot s_2&s_2\cdot(1-s_2)\end{bmatrix} \quad \frac{\partial J(\mathbf{l},\mathbf{s})}{\partial \mathbf{s}}=\begin{bmatrix}-I(l_i=1)\cdot\frac{1}{s_1}\\-I(l_i=2)\cdot\frac{1}{s_2}\end{bmatrix}}$$

Retropropagación en Redes Neuronales

x → Capa lineal → a() → Capa lineal → softmax → Cross-entropy

x	$z=\Omega_0 x+\beta_0$	$h=a(z)$	$y=\Omega_1 h+\beta_1$	s	J
3x1	3x1 3x3 3x1	3x1	2x3 2x1	2x1	1x1

$$s(y)=\begin{bmatrix}\frac{e^{y_1}}{e^{y_1}+e^{y_2}}\\ \frac{e^{y_2}}{e^{y_1}+e^{y_2}}\end{bmatrix}$$

$$J(l,s)=-\sum_{i=1}^{2} I(l_i=i)\log(s_i)$$

$$\frac{\partial J}{\partial x}=\underbrace{\underset{3x3}{\frac{\partial z}{\partial x}}\cdot\underbrace{\underset{3x3}{\frac{\partial h}{\partial z}}\cdot\underbrace{\underset{3x2}{\frac{\partial y}{\partial h}}\cdot\underbrace{\underset{2x2}{\frac{\partial s}{\partial y}}\cdot\underset{2x1}{\frac{\partial J}{\partial s}}}_{\frac{\partial J}{\partial y}\ 2x1}}_{\frac{\partial J}{\partial h}\ 3x1}}_{\frac{\partial J}{\partial z}\ 3x1}}_{\frac{\partial J}{\partial x}\ 3x1}$$

Multiplicar dos matrices nxn es $O(n^3)$, Multiplicar una matriz nxn por una nx1 es $O(n^2)$

Durante la retropropagación, se multiplican matrices por vectores → backprop es $O(n^2)$

Pregunta 5

x → Capa lineal → a() → Capa lineal → softmax → Cross-entropy

x	$z=\Omega_0 x+\beta_0$	$h=a(z)$	$y=\Omega_1 h+\beta_1$	s	J
3x1	3x1 3x3 3x1	3x1	2x3 2x1	2x1	1x1

$$s(y)=\begin{bmatrix}\frac{e^{y_1}}{e^{y_1}+e^{y_2}}\\ \frac{e^{y_2}}{e^{y_1}+e^{y_2}}\end{bmatrix}$$

$$J(l,s)=-\sum_{i=1}^{2} I(l_i=i)\log(s_i)$$

- ¿Qué complejidad tiene la evaluación de la red para una entrada (foward pass)? (suponiendo que las matrices en la red son nxn)