Scalar Operations

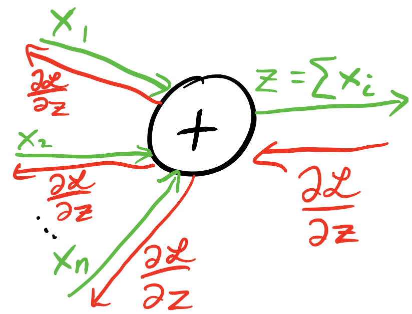
$+$

$$z = \sum_i x_i \Rightarrow \frac{\partial z}{\partial x_i} = 1$$

$$\mathcal{L} = f(z)$$

$$\frac{\partial \mathcal{L}}{\partial x_i} = \frac{\partial \mathcal{L}}{\partial z} \cdot \frac{\partial z}{\partial x_i} = \frac{\partial \mathcal{L}}{\partial z}$$

"distribute flow unchanged"



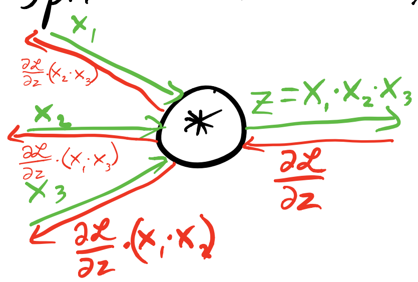
$*$

$$z = \prod_i x_i \Rightarrow \frac{\partial z}{\partial x_i} = \frac{z}{x_i}$$

$$\mathcal{L} = f(z)$$

$$\frac{\partial \mathcal{L}}{\partial x_i} = \frac{\partial \mathcal{L}}{\partial z} \cdot \frac{\partial z}{\partial x_i} = \frac{\partial \mathcal{L}}{\partial z} \cdot \frac{z}{x_i}$$

"split flow & scale by $\frac{out}{in_i}$"



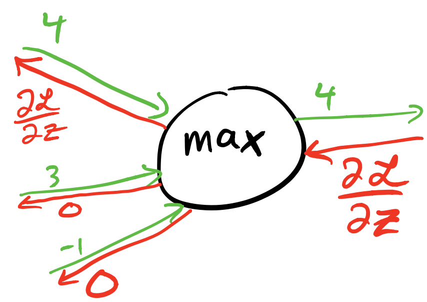
max & min

$$z = \max(x_1, x_2, \ldots, x_n) \Rightarrow \frac{\partial z}{\partial x_i} = \mathbb{1}[z == x_i]$$

$$\mathcal{L} = f(z)$$

$$\frac{\partial \mathcal{L}}{\partial x_i} = \frac{\partial \mathcal{L}}{\partial z} \, \mathbb{1}[x_i == z]$$

"direct flow to max/min input"



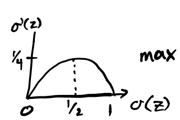
$\sigma = \frac{1}{1+e^{-x}}$

$$z = \sigma(x) \Rightarrow \frac{\partial z}{\partial x} = z(1-z)$$

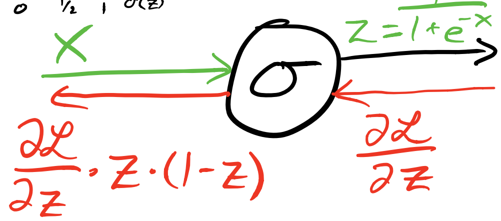
$$\mathcal{L} = f(z)$$

$$\frac{\partial \mathcal{L}}{\partial x} = \frac{\partial \mathcal{L}}{\partial z} \cdot z \cdot (1-z)$$

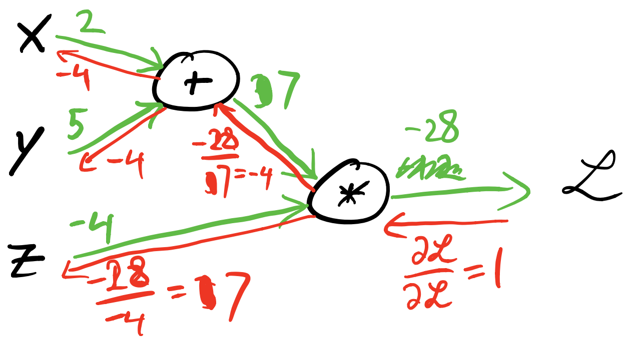
"attenuate flow increasingly as output saturates"



max is $\frac{1}{4} \Rightarrow$ vanishing gradient

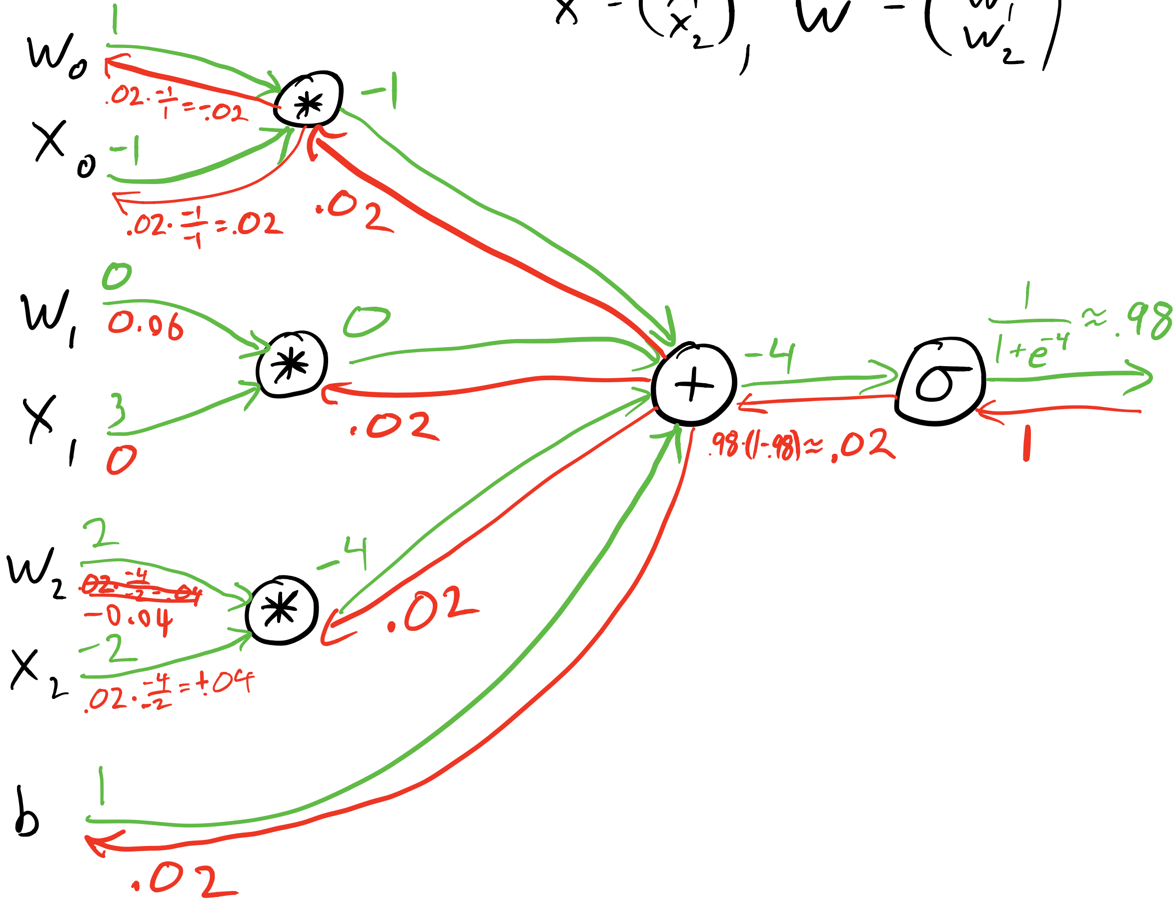


1. $\mathcal{L} = (x+y)z$



2. $\mathcal{L} = \sigma(\vec{w}^T\vec{x} + b)$ $(\vec{x}, \vec{w} \in \mathbb{R}^{3\times 1}, b \in \mathbb{R})$

$$\vec{x} = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix}, \quad \vec{w} = \begin{pmatrix} w_0 \\ w_1 \\ w_2 \end{pmatrix}$$

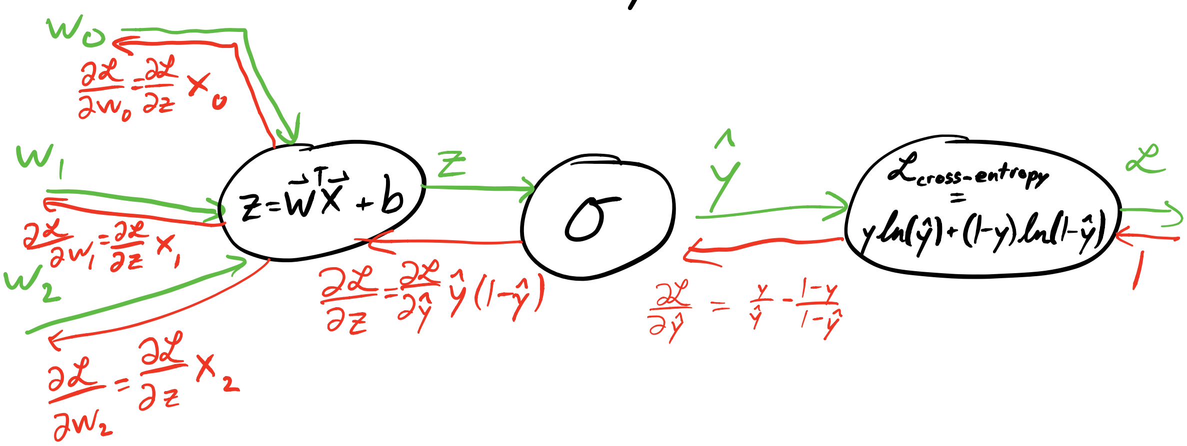


Logistic Regression (Batch Size = 1)

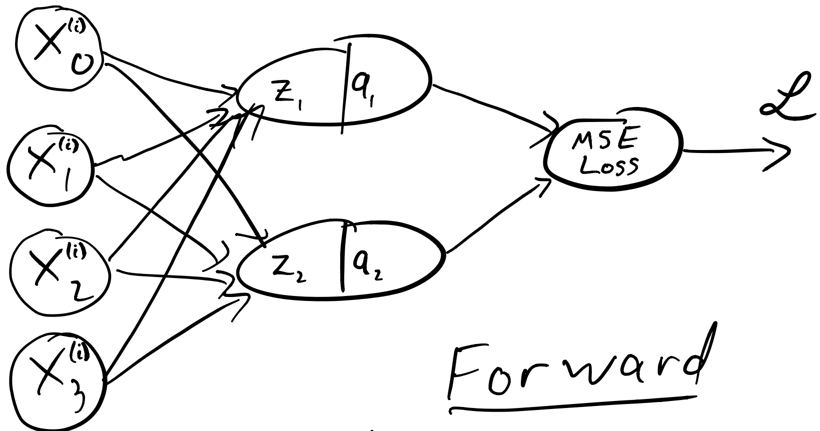
given: x_i inputs, y output

$$\vec{x} = \mathbb{R}^{n\times 1},\ \vec{w} \in \mathbb{R}^{n\times 1},\ b \in \mathbb{R}$$

Forward	Backward
$z = \vec{w}^T\vec{x} + b$	$\frac{\partial \mathcal{L}}{\partial \vec{w}} = \frac{\partial \mathcal{L}}{\partial z}\frac{\partial z}{\partial \vec{w}} = \left(\frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}}\right)\hat{y}(1-\hat{y})\vec{x}$; $\frac{\partial \mathcal{L}}{\partial b} = \frac{\partial \mathcal{L}}{\partial z}\frac{\partial z}{\partial b} = \left(\frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}}\right)\hat{y}(1-\hat{y})$
$\hat{y} = \sigma(z)$	$\frac{\partial \mathcal{L}}{\partial z} = \frac{\partial \mathcal{L}}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial z} = \left(\frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}}\right)\left(\hat{y}(1-\hat{y})\right)$
$\mathcal{L} = y\ln(\hat{y}) + (1-y)\ln(1-\hat{y})$	$\frac{\partial \mathcal{L}}{\partial \hat{y}} = \frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}}$



2-Layer MLP (Batch Size=3)



Forward

1. $Z = W_1 X + b_1 u$

$$\underset{[4\times 3]}{X} = \begin{bmatrix} | & | & | \\ \text{example } 1 & \text{example } 2 & \text{example } 3 \\ | & | & | \end{bmatrix} = \begin{bmatrix} \text{— feature } 1 \text{ —} \\ \text{— feature } 2 \text{ —} \\ \text{— feature } 3 \text{ —} \\ \text{— feature } 4 \text{ —} \end{bmatrix} = \begin{bmatrix} X_1^{(1)} & X_1^{(2)} & X_1^{(3)} \\ X_2^{(1)} & X_2^{(2)} & X_2^{(3)} \\ X_3^{(1)} & X_3^{(2)} & X_3^{(3)} \\ X_4^{(1)} & X_4^{(2)} & X_4^{(3)} \end{bmatrix}$$

$$\underset{[2\times 4]}{W_1} = \begin{bmatrix} | & | & | & | \\ \text{weights of feature } 1 & \text{weights of feature } 2 & \text{weights of feature } 3 & \text{weights of feature } 4 \\ | & | & | & | \end{bmatrix} = \begin{bmatrix} \text{— weights of upper neuron —} \\ \text{— weights of lower neuron —} \end{bmatrix}$$

$$\underset{[2\times 1]}{b_1} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \quad \underset{1\times 3}{u} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

$$\underset{[2\times 3]}{Z} = \begin{bmatrix} \text{— preactivation in upper neuron —} \\ \text{— preactivation in lower neuron —} \end{bmatrix}$$

$$= \begin{bmatrix} | & | & | \\ \text{example } 1 \text{ preactivations} & \text{example } 2 \text{ preactivations} & \text{example } 3 \text{ preactivations} \\ | & | & | \end{bmatrix}$$

2. $A = \sigma(Z)$

$$\underset{[2\times 3]}{A} = \begin{bmatrix} \text{— activation of upper neuron —} \\ \text{— activation of lower neuron —} \end{bmatrix}$$

$$= \begin{bmatrix} | & | & | \\ \text{example 1 activations} & \text{example 2 activations} & \text{example 3 activations} \\ | & | & | \end{bmatrix}$$

3. $\hat{y} = w_2 A + b_2 u$

$$\underset{[1\times 2]}{W_2} \begin{bmatrix} \text{coefficient of upper activation} & \text{coefficient of lower activation} \end{bmatrix}$$

$b_2 \in \mathbb{R}$, $\underset{1\times 3}{u} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$

$$\underset{[1\times 3]}{\hat{y}} = \begin{bmatrix} \text{example 1 prediction} & \text{example 2 prediction} & \text{example 3 prediction} \end{bmatrix}$$

4. $\mathcal{L} = \frac{1}{3} \|\hat{y} - y\|^2$

$\mathcal{L} \in \mathbb{R}$

$$\underset{[1\times 3]}{y} = \begin{bmatrix} \text{ex. 1 ground truth} & \text{ex. 2 ground truth} & \text{ex. 3 ground truth} \end{bmatrix}$$

Backward

1. $\dfrac{\partial \mathcal{L}}{\partial \hat{y}} = \dfrac{2}{3}(\hat{y} - y)$

$[1 \times 3]$

2. $\dfrac{\partial \mathcal{L}}{\partial w_2} = \dfrac{\partial \mathcal{L}}{\partial \hat{y}} \dfrac{\partial \hat{y}}{\partial w_2} = \dfrac{2}{3}(\hat{y} - y) A^T$

$[1 \times 2]$

$\dfrac{\partial \mathcal{L}}{\partial b_2} = \dfrac{\partial \mathcal{L}}{\partial \hat{y}} \dfrac{\partial \hat{y}}{\partial b^2} = \dfrac{2}{3}(\hat{y} - y) u^T$

$[1 \times 1]$

$\dfrac{\partial \mathcal{L}}{\partial A} = \dfrac{2}{3} w_2^T (\hat{y} - y)$

$[2 \times 3]$

3. $\dfrac{\partial \mathcal{L}}{\partial Z} = \dfrac{\partial \mathcal{L}}{\partial A} \odot \sigma'(Z) = \left(\dfrac{2}{3} w_2^T (\hat{y} - y)\right) \odot A \odot (1 - A)$

$[2 \times 3]$

4. $$\underset{[2\times 4]}{\frac{\partial \mathcal{L}}{\partial W_1}} = \frac{\partial \mathcal{L}}{\partial Z}\frac{\partial Z}{\partial W_1} = \frac{2}{3}\Big(\big(w_2^T(\hat{y}-y)\big)\odot A\odot(1-A)\Big)X^T$$

$$\underset{[2\times 1]}{\frac{\partial \mathcal{L}}{\partial b_1}} = \frac{\partial \mathcal{L}}{\partial Z}\frac{\partial Z}{\partial b_1} = \frac{2}{3}\Big(\big(w_2^T(\hat{y}-y)\big)\odot A\odot(1-A)\Big)u^T$$

In general...

$$Z^{[\ell]} = W^{[\ell]}A^{[\ell-1]} + b^{[\ell]}$$

$$A^{[\ell]} = g^{[\ell]}(Z^{[\ell]})$$ (where g is an elementwise nonlinearity like σ or ReLU)

Let $\delta^{[\ell]} = \frac{\partial \mathcal{L}}{\partial Z^{[\ell]}}$, which we can calculate starting at the last layer using $\delta^{[N]} = \frac{\partial \mathcal{L}}{\partial \hat{y}} \odot g'^{[N]}(Z^{[N]})$ and working backwards either by

$$\delta^{[\ell]} = \left(\left(W^{[\ell+1]}\right)^T\delta^{[\ell+1]}\right)\odot g'^{[\ell]}(Z^{[\ell]}).$$

Then $$\frac{\partial \mathcal{L}}{\partial W^{[\ell]}} = \delta^{[\ell]}\left(A^{[\ell-1]}\right)^T$$

$$\frac{\partial \mathcal{L}}{\partial b^{[\ell]}} = \delta^{[\ell]}u^T$$

In our example we had
$g^{[1]} = \sigma$ and $g^{[2]} = $ identity function

[2]

$$\delta^{[2]} = \frac{\partial \mathcal{L}}{\partial \hat{y}} = \frac{2}{3}(\hat{y}-y), \quad \frac{\partial \mathcal{L}}{\partial W^{[2]}} = \frac{2}{3}(\hat{y}-y)A^T, \quad \frac{\partial \mathcal{L}}{\partial b^{[2]}} = \frac{2}{3}(\hat{y}-y)u^T$$

[1]

$$\delta^{[1]} = \left(\left(W^{[2]}\right)^T \cdot \frac{2}{3}(\hat{y}-y)\right) \odot A \odot (1-A)$$

$$\frac{\partial \mathcal{L}}{\partial W^{[1]}} = \delta^{[1]}\left(A^{[0]}\right)^T = \left(\left(\left(W^{[2]}\right)^T \cdot \frac{2}{3}(\hat{y}-y)\right) \odot A \odot (1-A)\right) X^T$$

$$\frac{\partial \mathcal{L}}{\partial b^{[1]}} = \delta^{[1]}\left(A^{[0]}\right)^T = \left(\left(\left(W^{[2]}\right)^T \cdot \frac{2}{3}(\hat{y}-y)\right) \odot A \odot (1-A)\right) u^T$$

✓

Optimizers

- Mini-Batch Gradient Descent:

"learning rate"

$$W^{[\ell]} = W^{[\ell]} - \alpha \frac{\partial \mathcal{L}}{\partial W^{[\ell]}}$$

$$b^{[\ell]} = b^{[\ell]} - \alpha \frac{\partial \mathcal{L}}{\partial b^{[\ell]}}$$

(Stochastic Gradient Descent has batch size = 1)

- Momentum:

  Use exponential weighted average of past gradient updates rather than the raw ones (dampens oscillations)

$$V^{[\ell]} = \beta V^{[\ell]} + (1-\beta) \frac{\partial \mathcal{L}}{\partial W^{[\ell]}}$$

$$W^{[\ell]} = W^{[\ell]} - \alpha V^{[\ell]}$$

- RMSProp:

  Divide gradient by exponential weighted average of its magnitude

$$S^{[\ell]} = \beta S^{[\ell]} + (1-\beta) \left(\frac{\partial \mathcal{L}}{\partial W^{[\ell]}} \right)^{\odot 2}$$ (element-wise square)

$$W^{[\ell]} = W^{[\ell]} - \alpha \frac{\partial \mathcal{L} / \partial W^{[\ell]}}{\sqrt{S^{[\ell]}} + \varepsilon}$$ (ε = small non-zero param to avoid dividing by 0)

- Adam:

Combine Momentum & RMS Prop with an additional bias correction

$$V = \frac{\beta_1 V + (1-\beta_1)\frac{\partial \mathcal{L}}{\partial W}}{(1-\beta_1)^t}$$

(dividing by $(1-\beta)^t$ yields less biased values for small t)

$$S = \frac{\beta_2 S + (1-\beta_2)\left(\frac{\partial \mathcal{L}}{\partial W}\right)^{\odot 2}}{(1-\beta_2)^t}$$

$$W = W - \alpha \frac{V}{\sqrt{S} + \varepsilon}$$