Backpropagation

Gradient Descent

Network parameters $\theta = \{w_1, w_2, \cdots, b_1, b_2, \cdots\}$

Starting Parameters $\theta^0 \longrightarrow \theta^1 \longrightarrow \theta^2 \longrightarrow \cdots\cdots$

$$\nabla L(\theta) = \begin{bmatrix} \partial L(\theta)/\partial w_1 \\ \partial L(\theta)/\partial w_2 \\ \vdots \\ \partial L(\theta)/\partial b_1 \\ \partial L(\theta)/\partial b_2 \\ \vdots \end{bmatrix}$$

$Compute\ \nabla L(\theta^0)$ $\quad \theta^1 = \theta^0 - \eta \nabla L(\theta^0)$

$Compute\ \nabla L(\theta^1)$ $\quad \theta^2 = \theta^1 - \eta \nabla L(\theta^1)$

Millions of parameters ……

To compute the gradients efficiently, we use ***backpropagation***.

Chain Rule

Case 1

$$y = g(x) \quad z = h(y)$$

$$\Delta x \to \Delta y \to \Delta z \qquad \frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx}$$

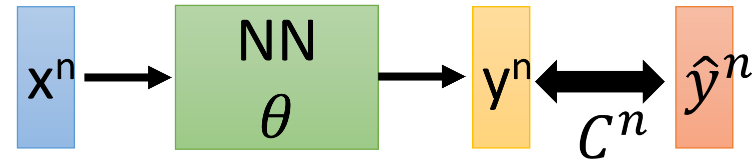
Case 2

$$x = g(s) \qquad y = h(s) \qquad z = k(x, y)$$

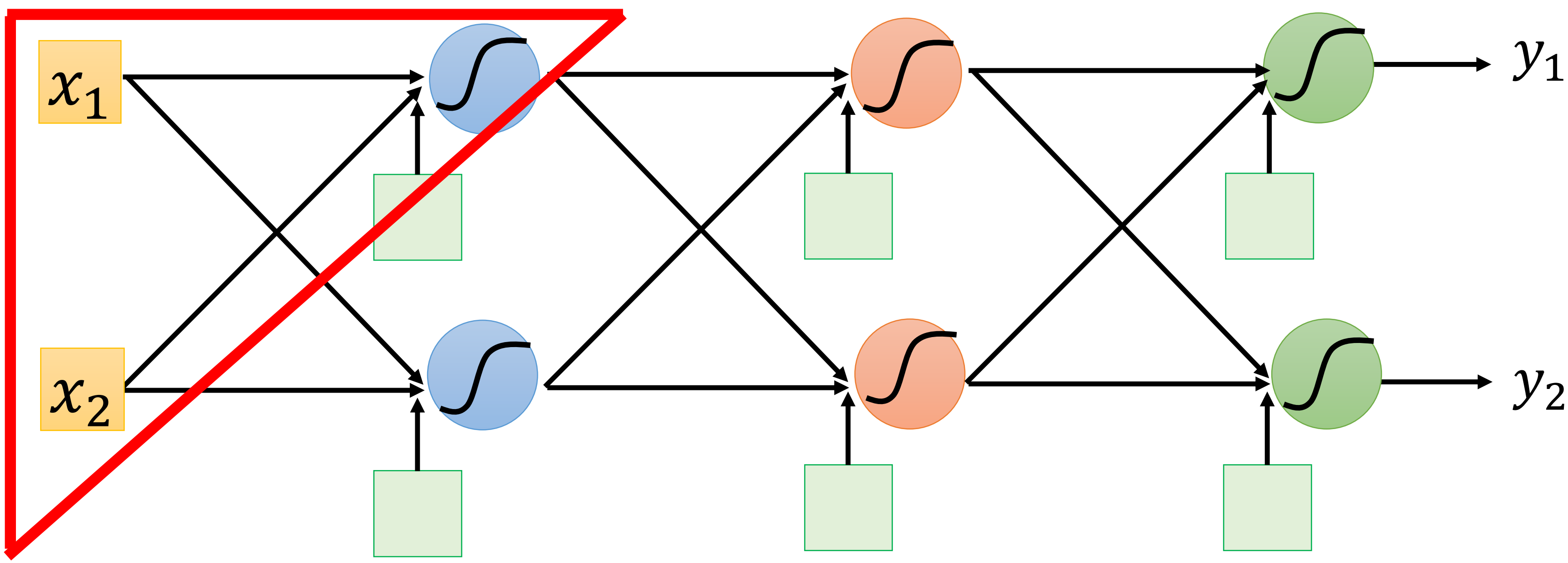
$$\begin{tikzcd}
 & \Delta x \arrow[dr] & \\
\Delta s \arrow[ur] \arrow[dr] & & \Delta z \\
 & \Delta y \arrow[ur] &
\end{tikzcd}$$

$$\frac{dz}{ds} = \frac{\partial z}{\partial x}\frac{dx}{ds} + \frac{\partial z}{\partial y}\frac{dy}{ds}$$

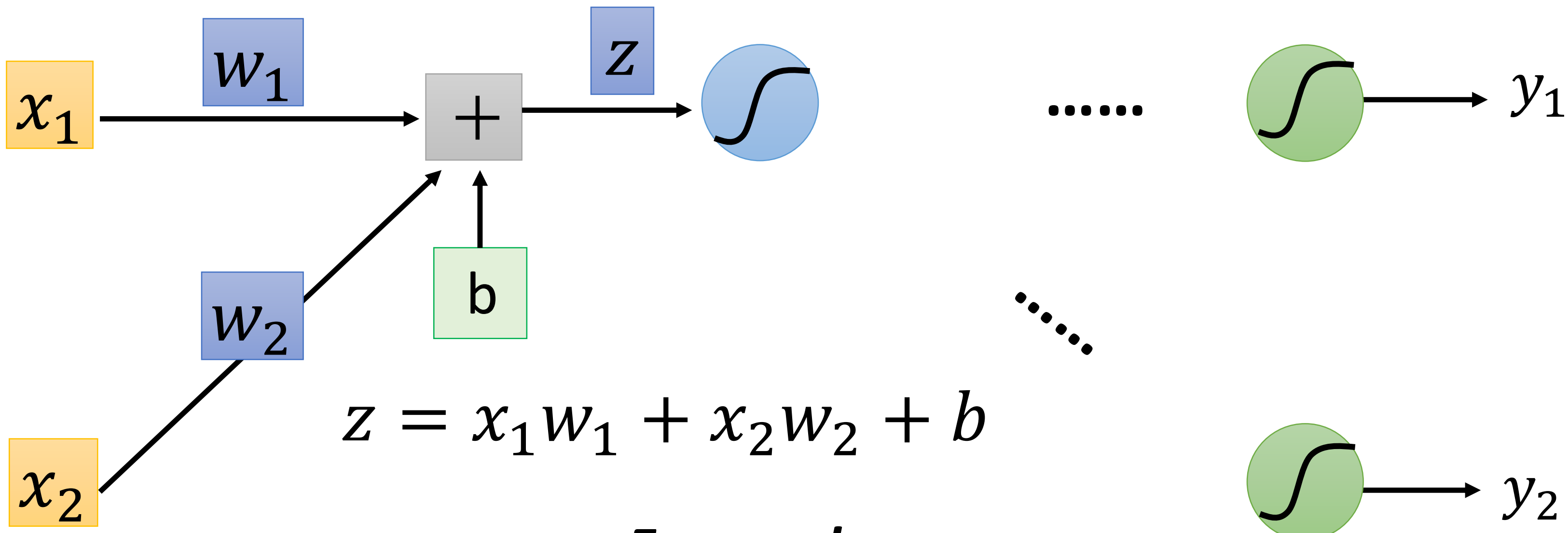
Backpropagation



$$L(\theta) = \sum_{n=1}^{N} C^n(\theta) \quad \Rightarrow \quad \frac{\partial L(\theta)}{\partial w} = \sum_{n=1}^{N} \frac{\partial C^n(\theta)}{\partial w}$$



Backpropagation



$$z = x_1 w_1 + x_2 w_2 + b$$

$$\frac{\partial C}{\partial w} = ? \quad \frac{\partial z}{\partial w}\frac{\partial C}{\partial z}$$

(Chain rule)

<u>Forward pass:</u>

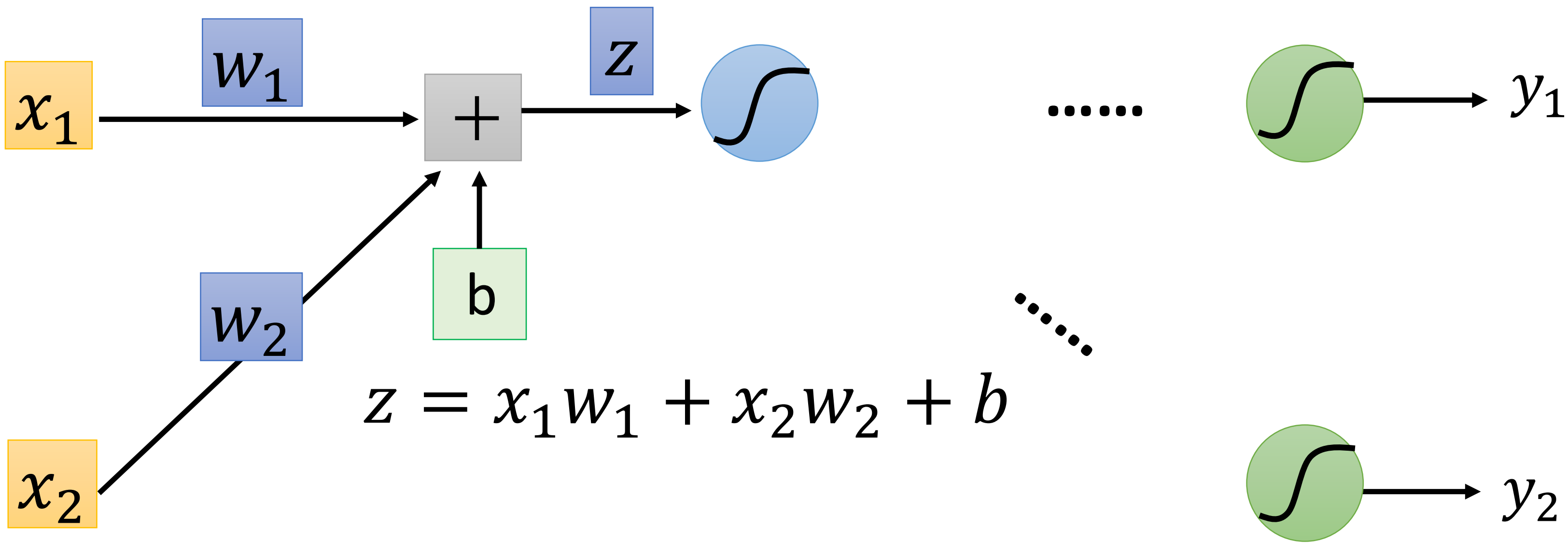
Compute $\partial z / \partial w$ for all parameters

<u>Backward pass:</u>

Compute $\partial C / \partial z$ for all activation function inputs z

Backpropagation – Forward pass

Compute $\partial z / \partial w$ for all parameters



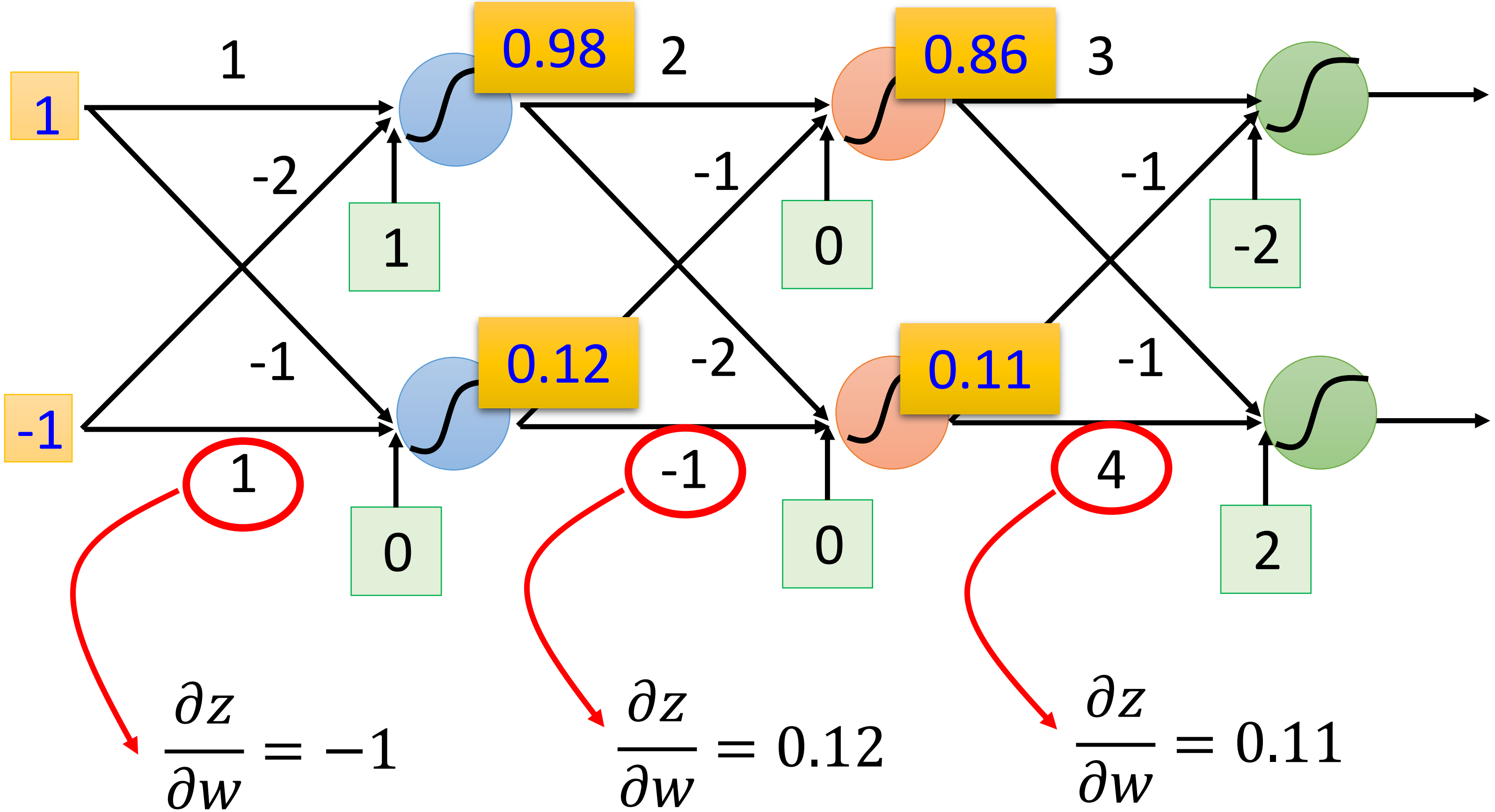
$$z = x_1 w_1 + x_2 w_2 + b$$

$$\left.\begin{array}{l} \partial z / \partial w_1 = ? \;\; x_1 \\ \partial z / \partial w_2 = ? \;\; x_2 \end{array}\right\}$$

The value of the input connected by the weight

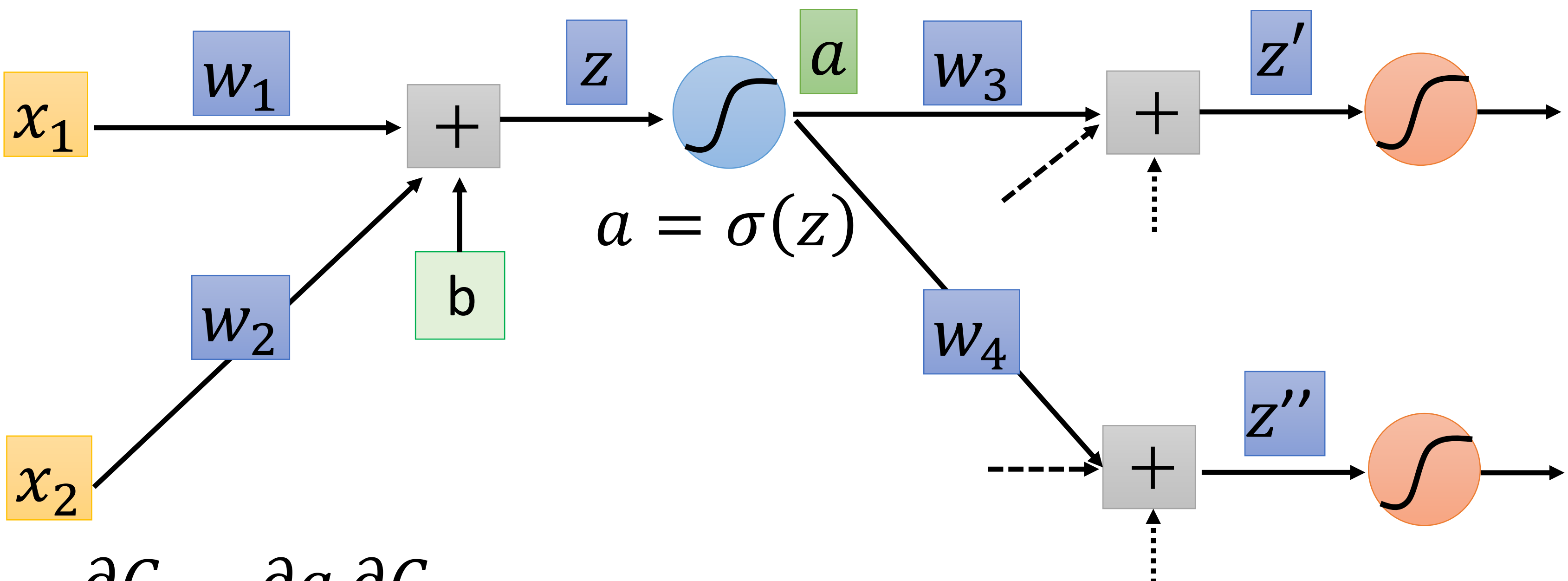
Backpropagation – Forward pass

Compute $\partial z/\partial w$ for all parameters



Backpropagation – Backward pass

Compute $\partial C / \partial z$ for all activation function inputs z

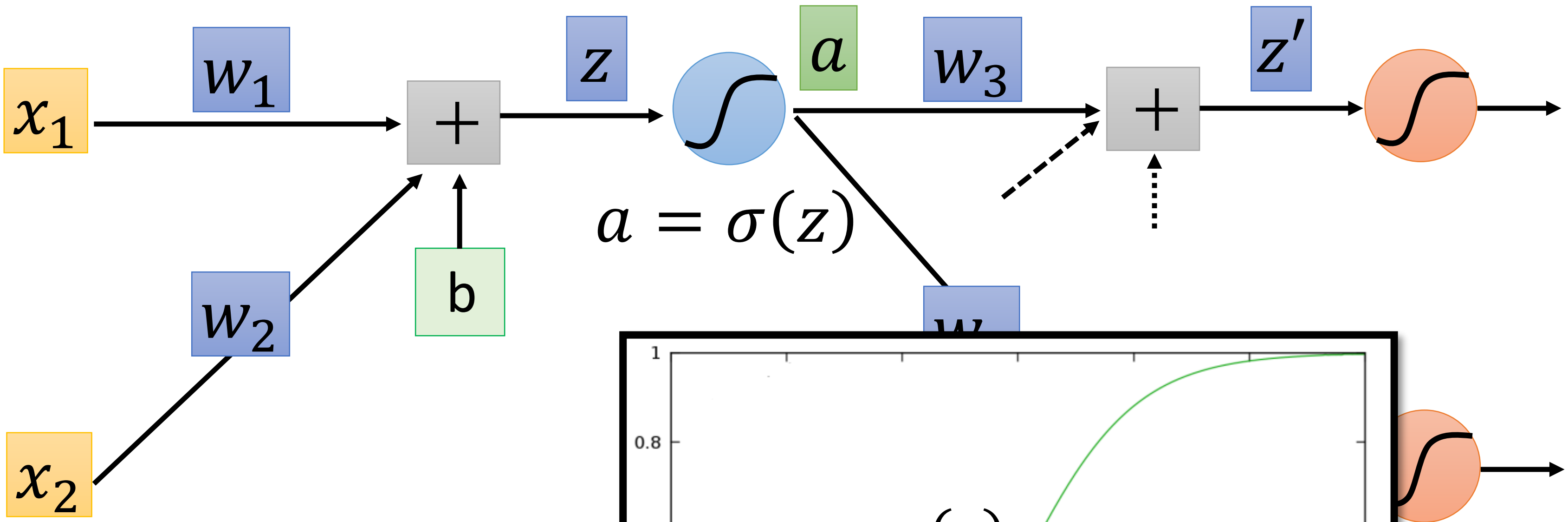


$$\frac{\partial C}{\partial z} = \underline{\frac{\partial a}{\partial z}} \frac{\partial C}{\partial a}$$

$\sigma'(z)$

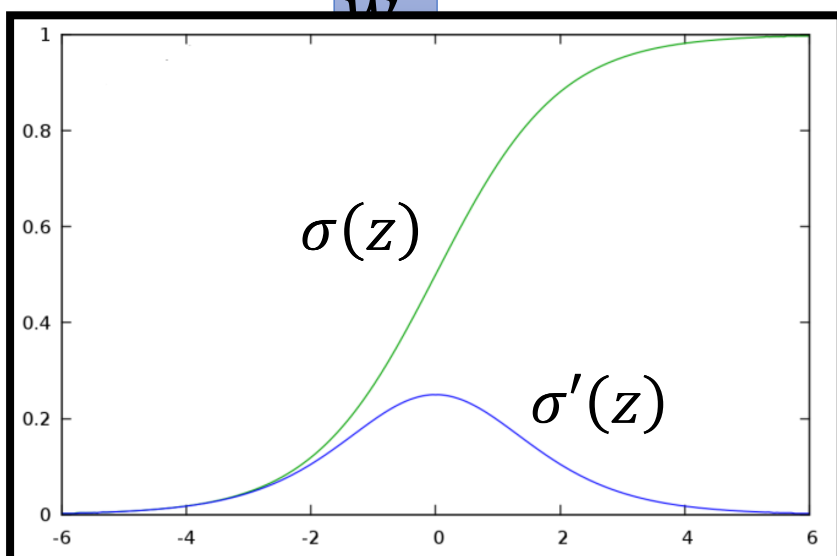
Backpropagation – Backward pass

Compute $\partial C / \partial z$ for all activation function inputs z



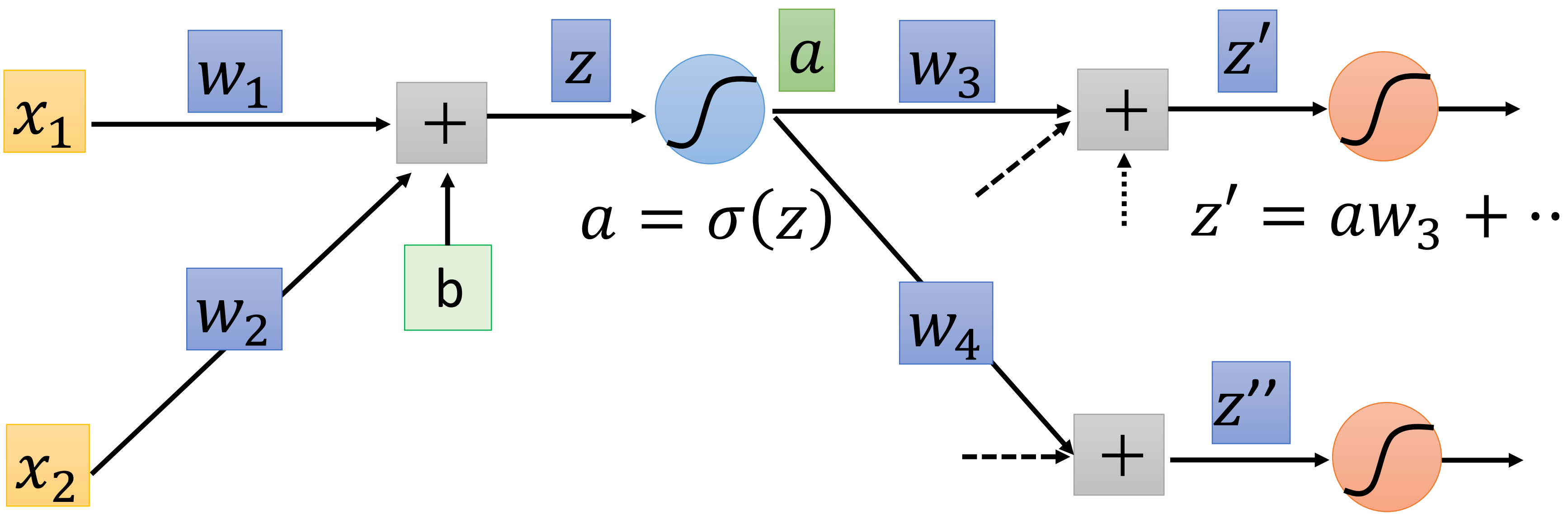
$$\frac{\partial C}{\partial z} = \frac{\partial a}{\partial z}\frac{\partial C}{\partial a}$$

$\rightarrow \sigma'(z)$



Backpropagation – Backward pass

Compute $\partial C/\partial z$ for all activation function inputs z



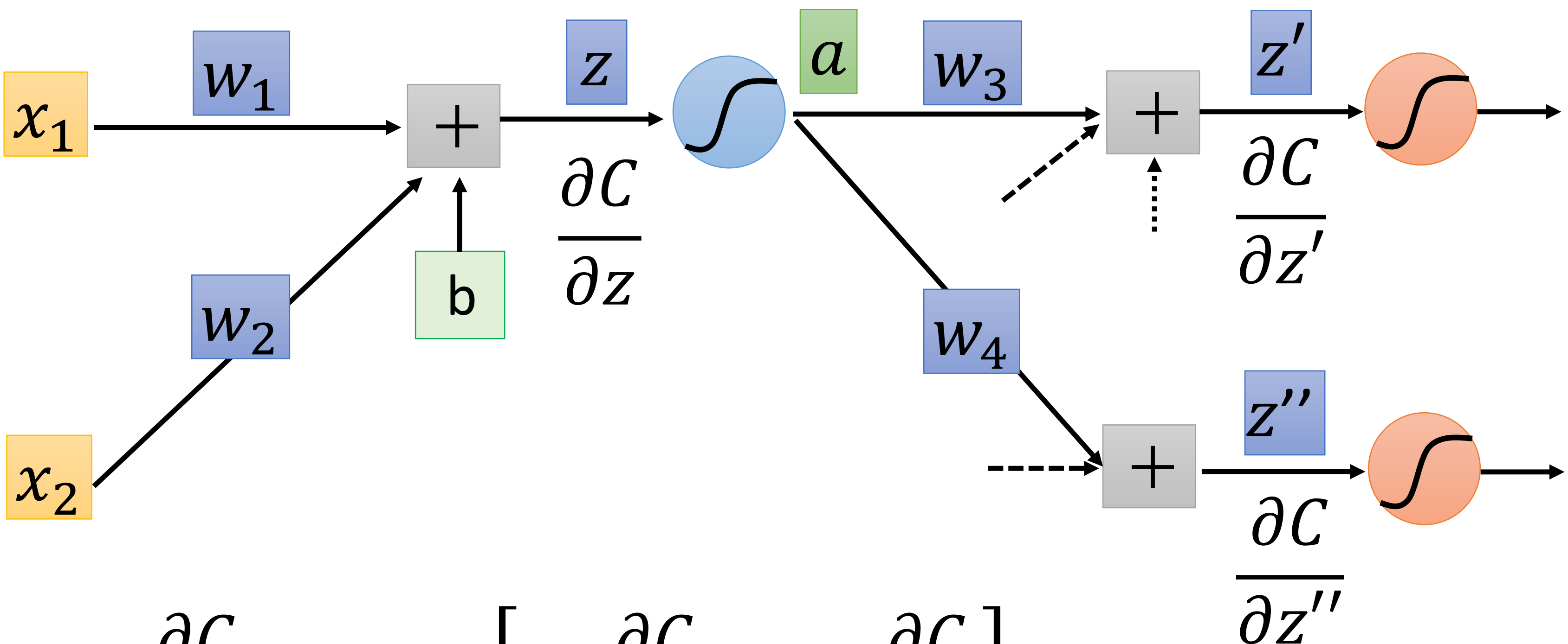
$$\frac{\partial C}{\partial z}=\frac{\partial a}{\partial z}\underbrace{\frac{\partial C}{\partial a}}$$

$$\frac{\partial C}{\partial a}=\underbrace{\frac{\partial z'}{\partial a}}_{w_3}\underbrace{\frac{\partial C}{\partial z'}}_{?}+\underbrace{\frac{\partial z''}{\partial a}}_{w_4}\underbrace{\frac{\partial C}{\partial z''}}_{?}$$ (Chain rule)

Assumed it's known

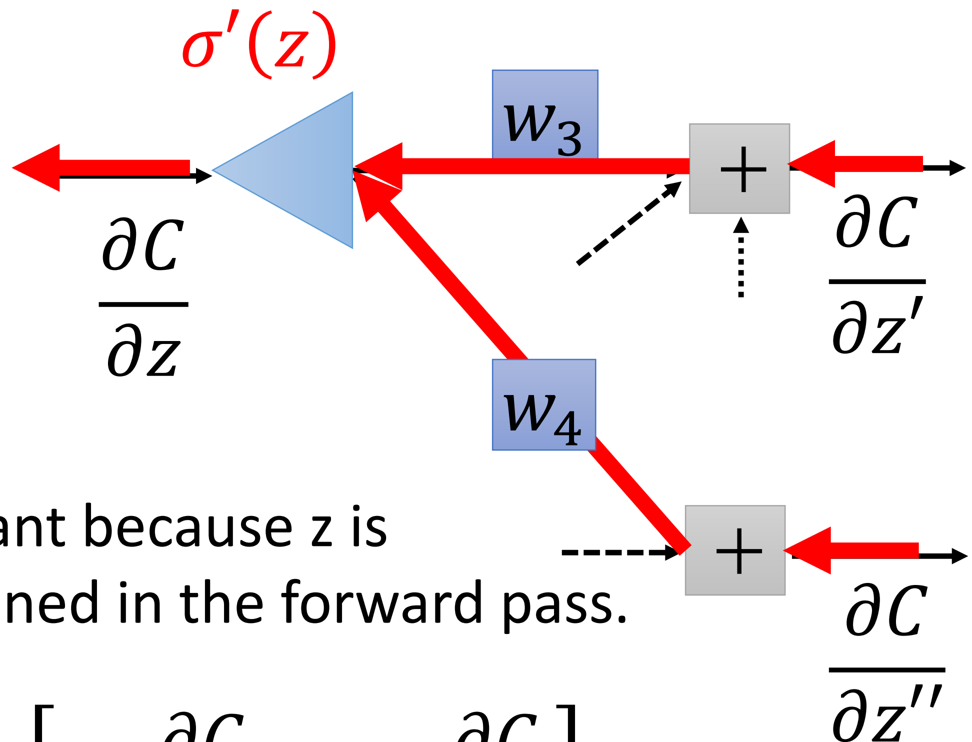
Backpropagation – Backward pass

Compute $\partial C / \partial z$ for all activation function inputs z



$$\frac{\partial C}{\partial z} = \sigma'(z)\left[w_3 \frac{\partial C}{\partial z'} + w_4 \frac{\partial C}{\partial z''}\right]$$

Backpropagation – Backward pass

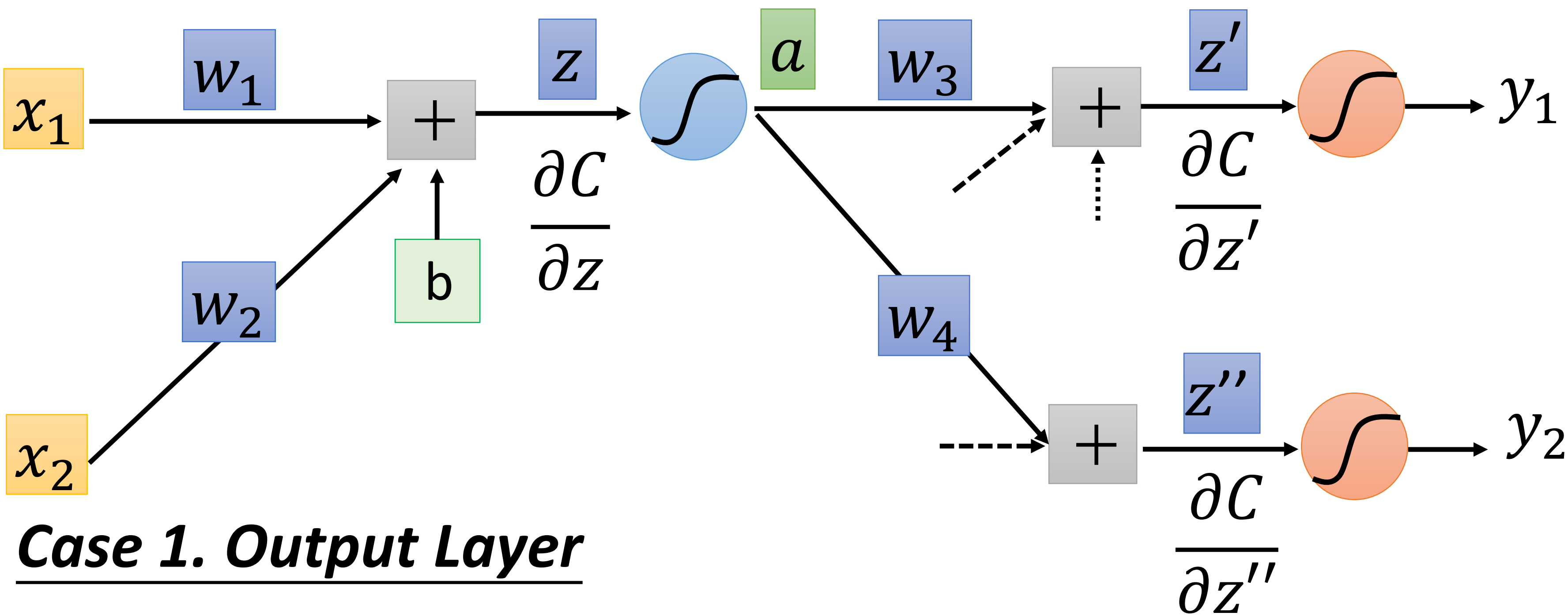


$\sigma'(z)$ is a constant because z is already determined in the forward pass.

$$\frac{\partial C}{\partial z} = \sigma'(z)\left[w_3 \frac{\partial C}{\partial z'} + w_4 \frac{\partial C}{\partial z''}\right]$$

Backpropagation – Backward pass

Compute $\partial C / \partial z$ for all activation function inputs z



<u>Case 1. Output Layer</u>

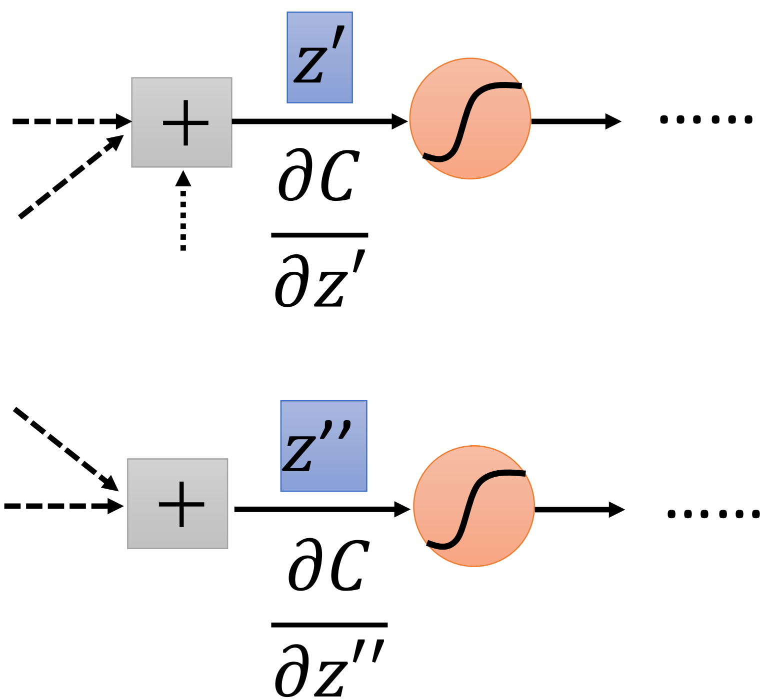
$$\frac{\partial C}{\partial z'} = \frac{\partial y_1}{\partial z'} \frac{\partial C}{\partial y_1} \qquad \frac{\partial C}{\partial z''} = \frac{\partial y_2}{\partial z''} \frac{\partial C}{\partial y_2}$$

Done!

Backpropagation – Backward pass

Compute $\partial C / \partial z$ for all activation function inputs z

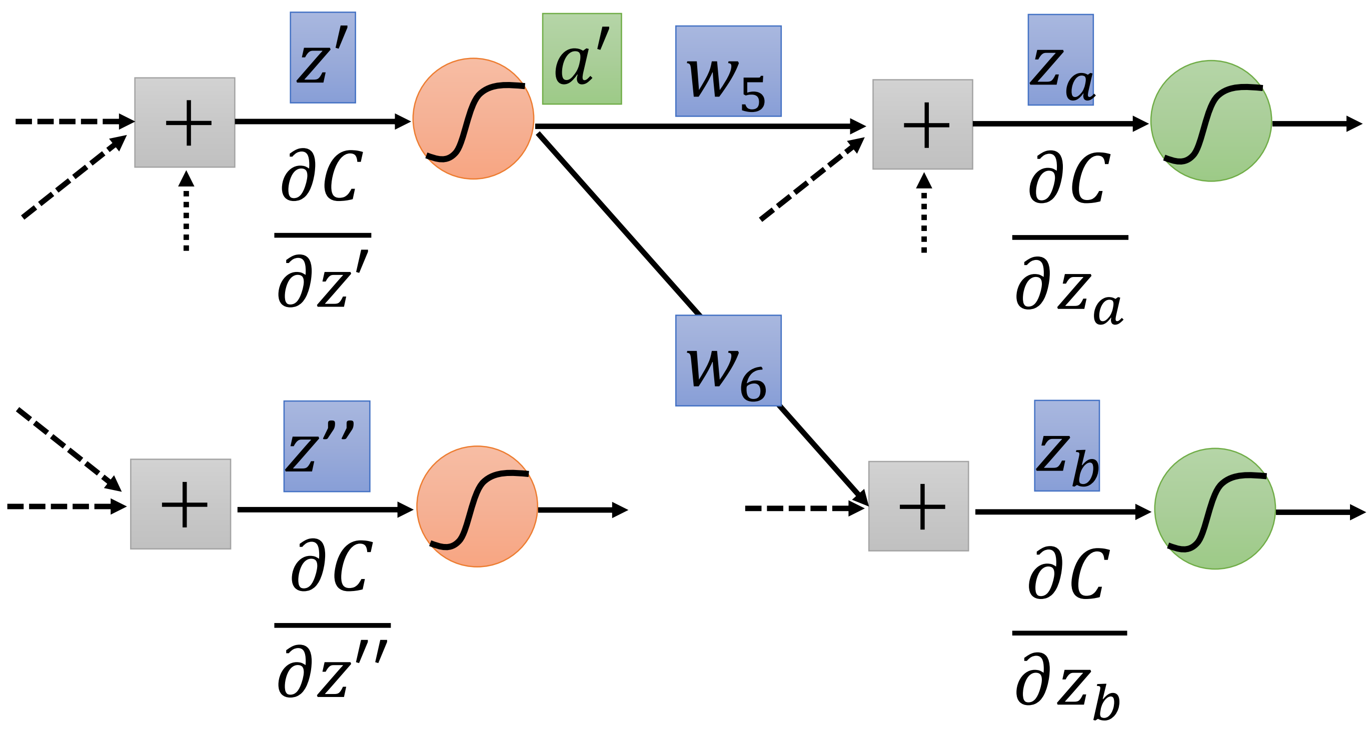
<u>Case 2. Not Output Layer</u>



Backpropagation – Backward pass

Compute $\partial C/\partial z$ for all activation function inputs z

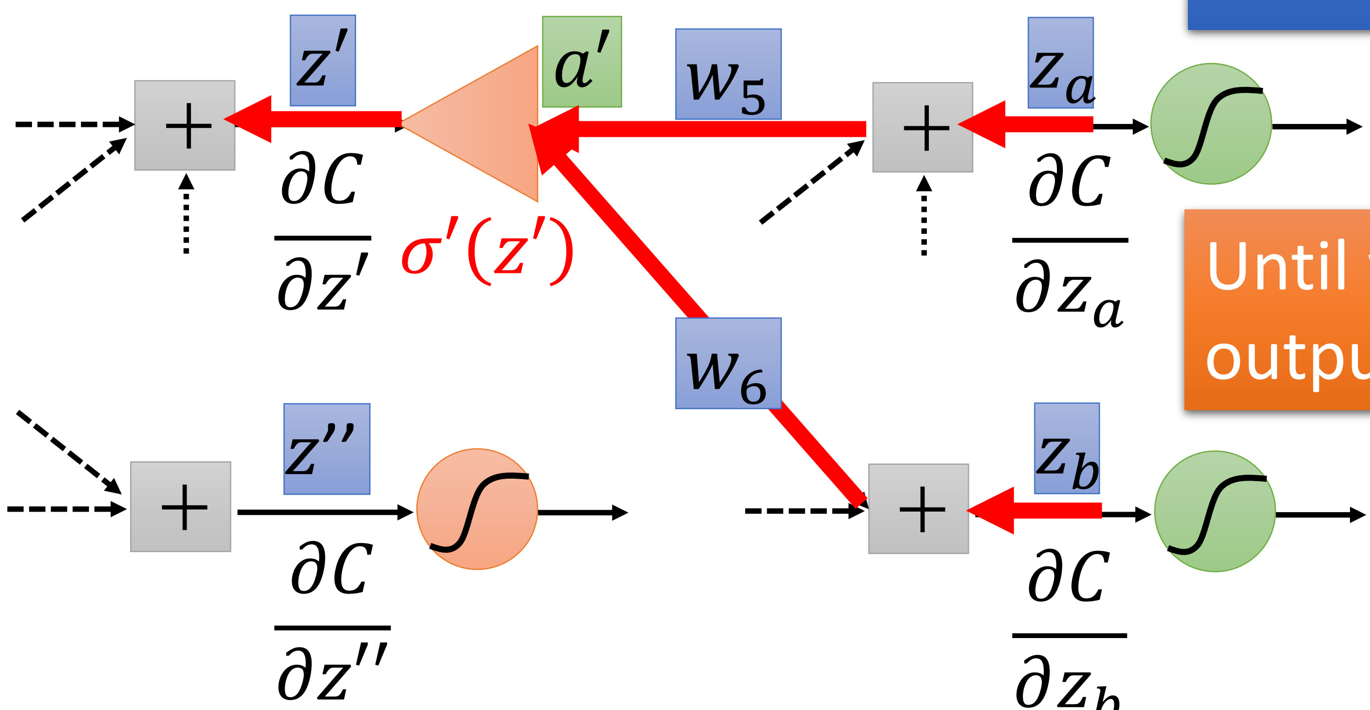
Case 2. Not Output Layer



Backpropagation – Backward pass

Compute $\partial C / \partial z$ for all activation function inputs z

Case 2. Not Output Layer



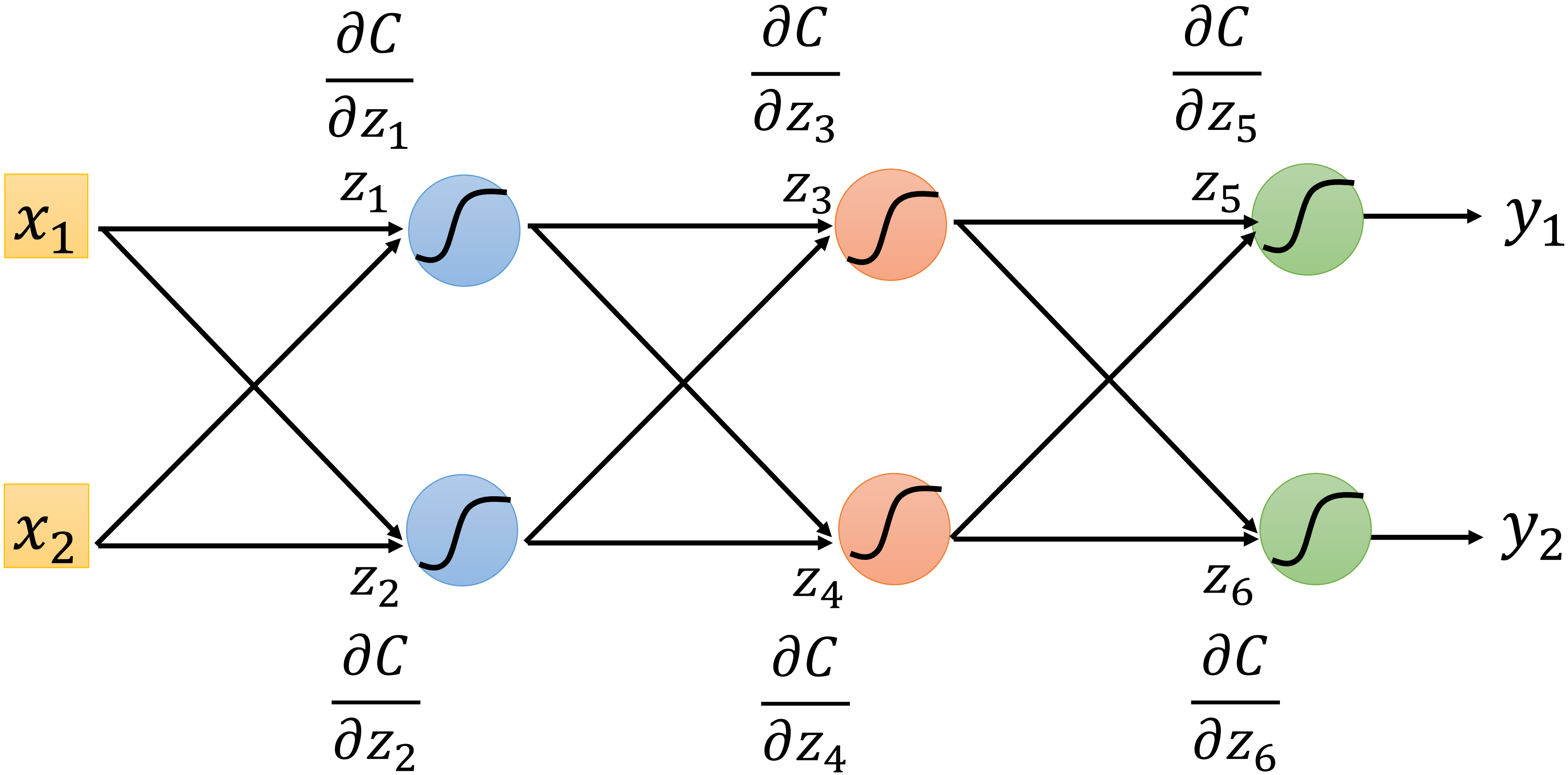
Compute $\partial C / \partial z$ recursively

Until we reach the output layer ……

Backpropagation – Backward Pass

Compute $\partial C / \partial z$ for all activation function inputs z

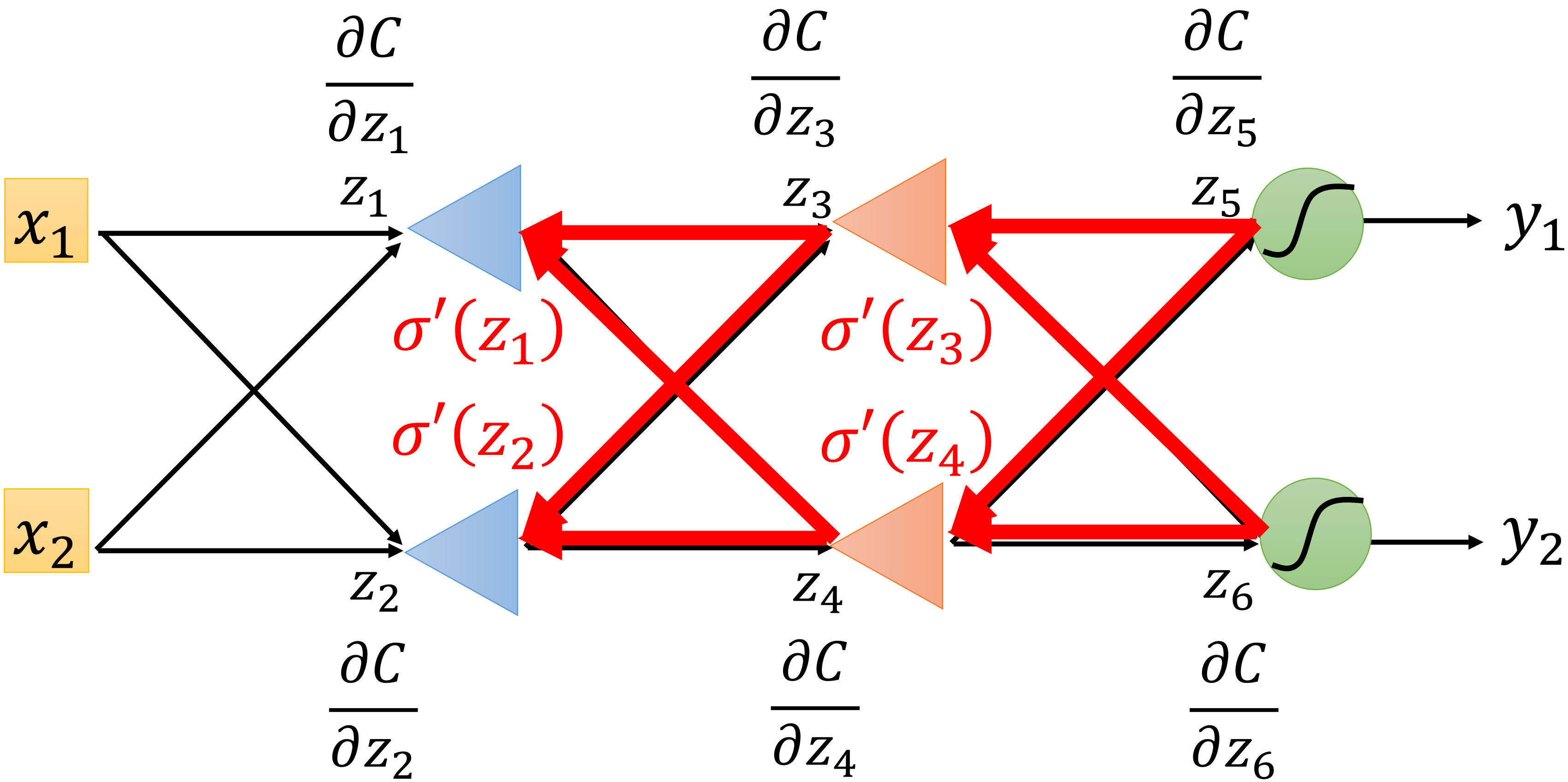
Compute $\partial C / \partial z$ from the output layer



Backpropagation – Backward Pass

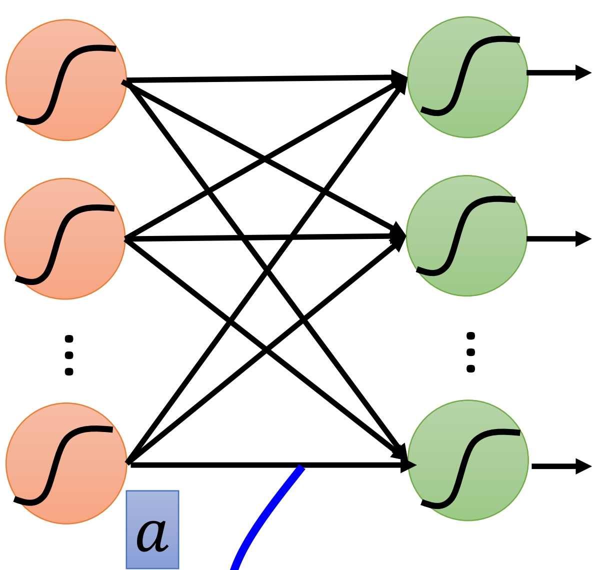
Compute $\partial C / \partial z$ for all activation function inputs z

Compute $\partial C / \partial z$ from the output layer

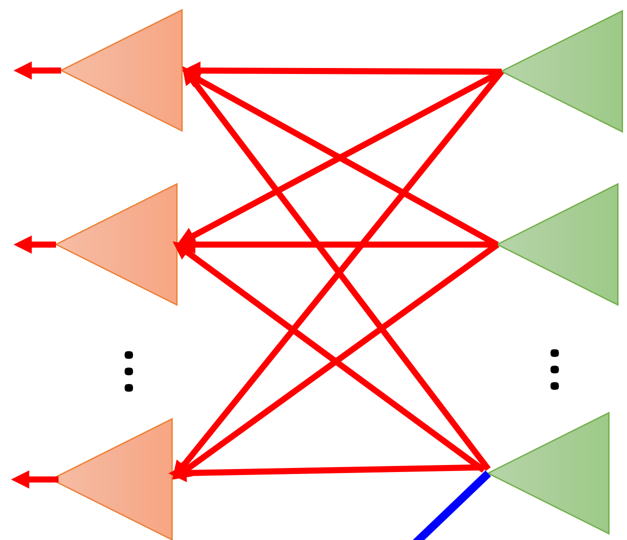


Backpropagation – Summary

Forward Pass



Backward Pass



$$\frac{\partial z}{\partial w} = a \quad \times \quad \frac{\partial C}{\partial z} \quad = \frac{\partial C}{\partial w}$$

for all w

反向传播中

第一级残差 $\delta^L = \frac{\partial C}{\partial a} \odot \sigma'(z^L) = \frac{\partial C}{\partial z^L}$

往后残差每一级乘权重传播 $\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l) = \frac{\partial C}{\partial z^l}$

$\frac{\partial C}{\partial b_j^l} = \delta_j^l$ $\quad \frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l$ $\quad \underset{100\times N}{a^{l-1}}$ $\quad \underset{10\times N}{z^l}$

$\underset{10\times 100}{w^l}$

矩阵形式：

$a = \sigma(z)$, $z = wx + b$

$\Rightarrow \frac{\partial a}{\partial x} = w^T \frac{\partial a}{\partial z}$, $\frac{\partial a}{\partial w} = \frac{\partial a}{\partial z} \cdot x^T$

$\therefore \frac{\partial C}{\partial w^l} = \delta^l \cdot a^{l-1\,T}$ $\quad \frac{\partial C}{\partial a^{l-1}} = w^{l\,T} \cdot \delta^l$ $\quad \frac{\partial C}{\partial b^{l-1}} = \mathrm{sum}(\delta^l, \mathrm{axis}=1)$