

Backpropagation

Hung-yi Lee

李宏毅

Gradient Descent

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

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

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

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

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

Millions of parameters

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

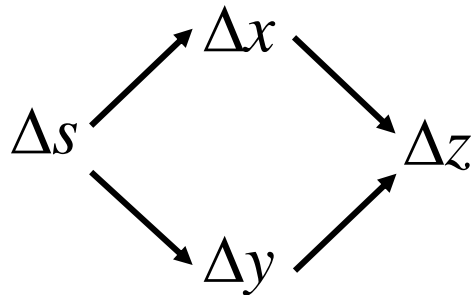
Chain Rule

Case 1 $y = g(x) \quad z = h(y)$

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

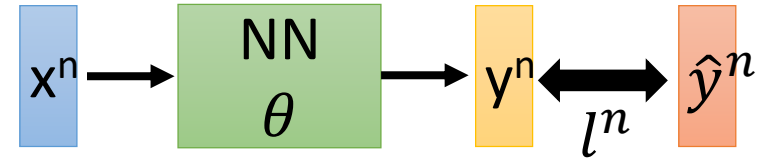
Case 2

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

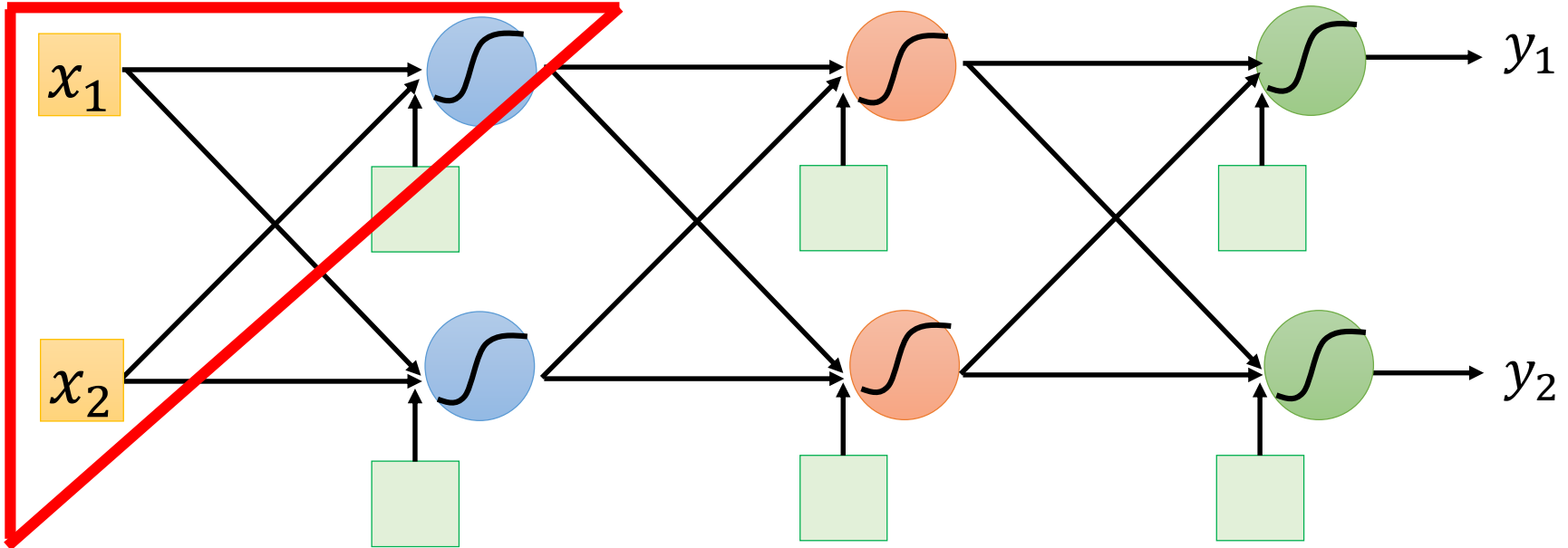


$$\frac{dz}{ds} = \frac{\partial z}{\partial x} \frac{dx}{ds} + \frac{\partial z}{\partial y} \frac{dy}{ds}$$

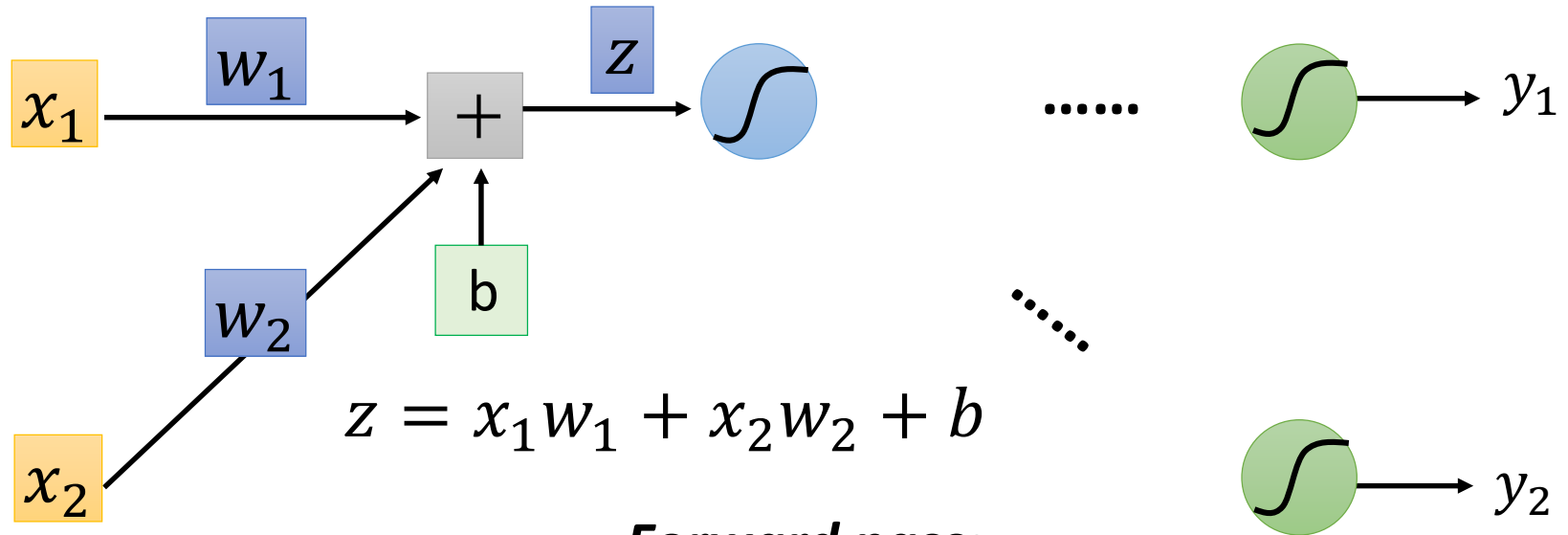
Backpropagation



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



Backpropagation



Forward pass:

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

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

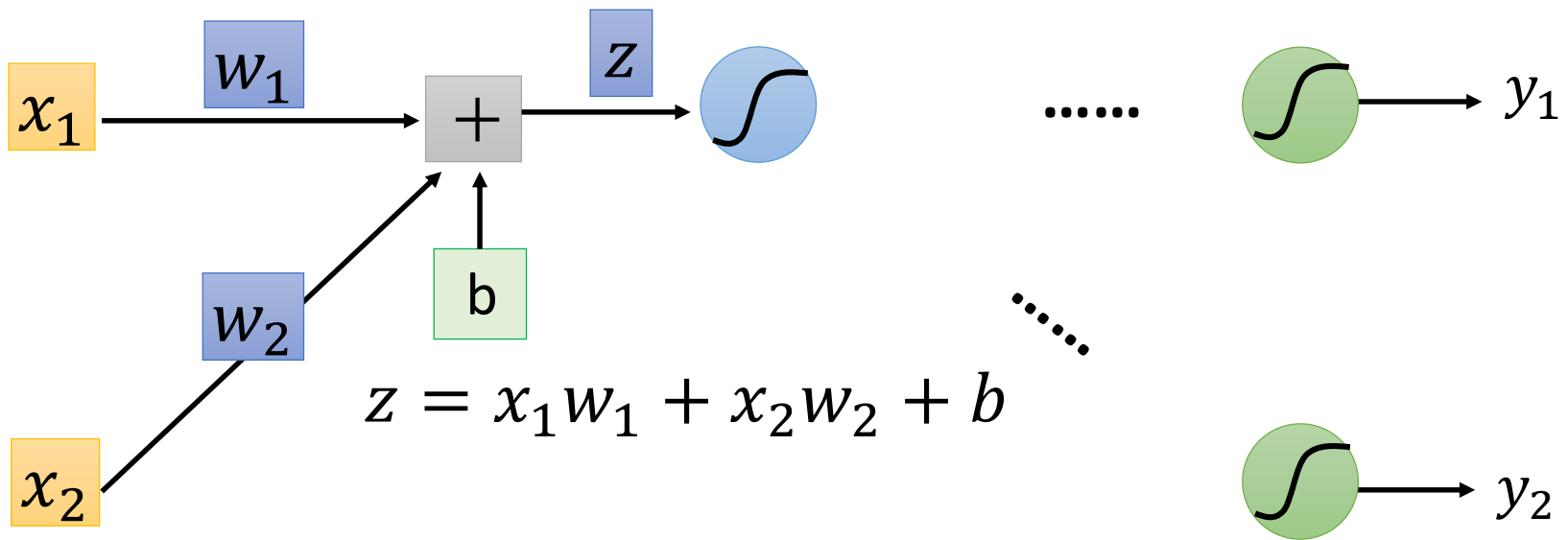
(Chain rule)

Backward pass:

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

Backpropagation – Forward pass

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



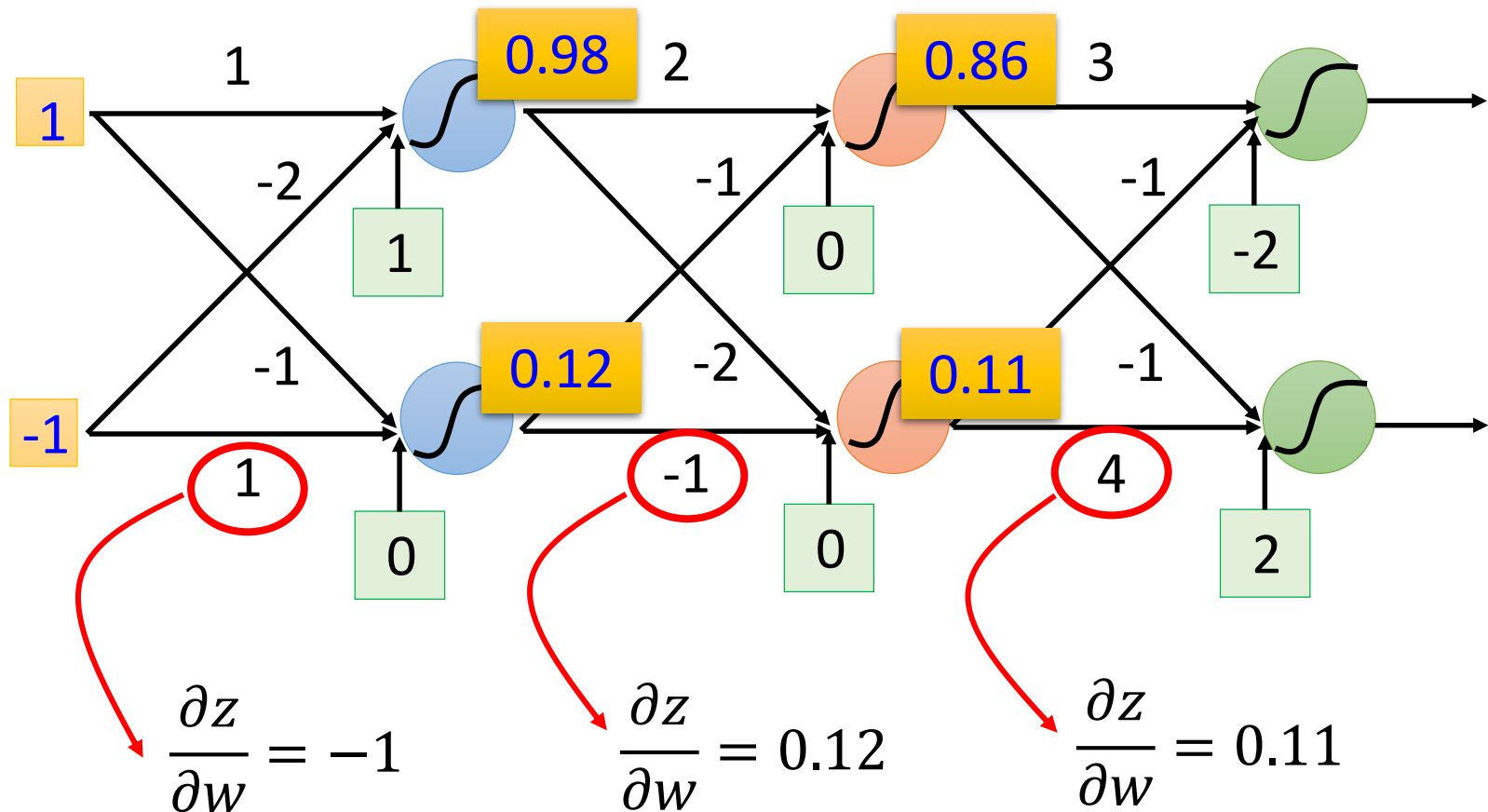
$$\partial z / \partial w_1 = ? \quad x_1$$

$$\partial z / \partial w_2 = ? \quad x_2$$

} The value of the input
connected by the weight

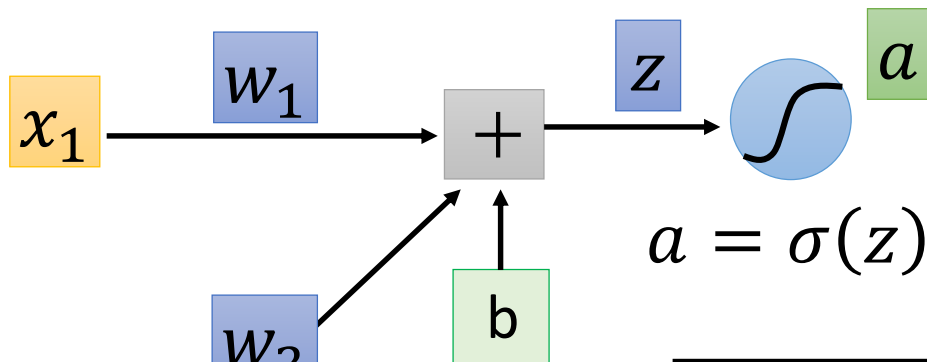
Backpropagation – Forward pass

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



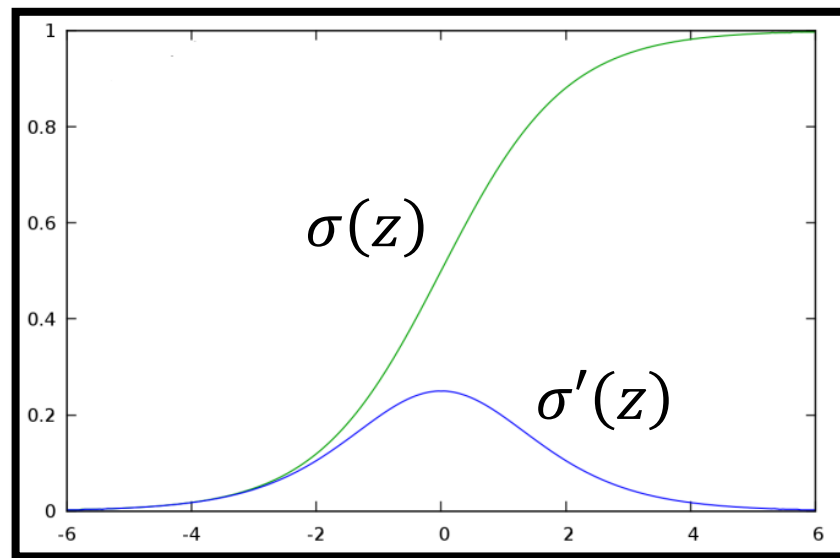
Backpropagation – Backward pass

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



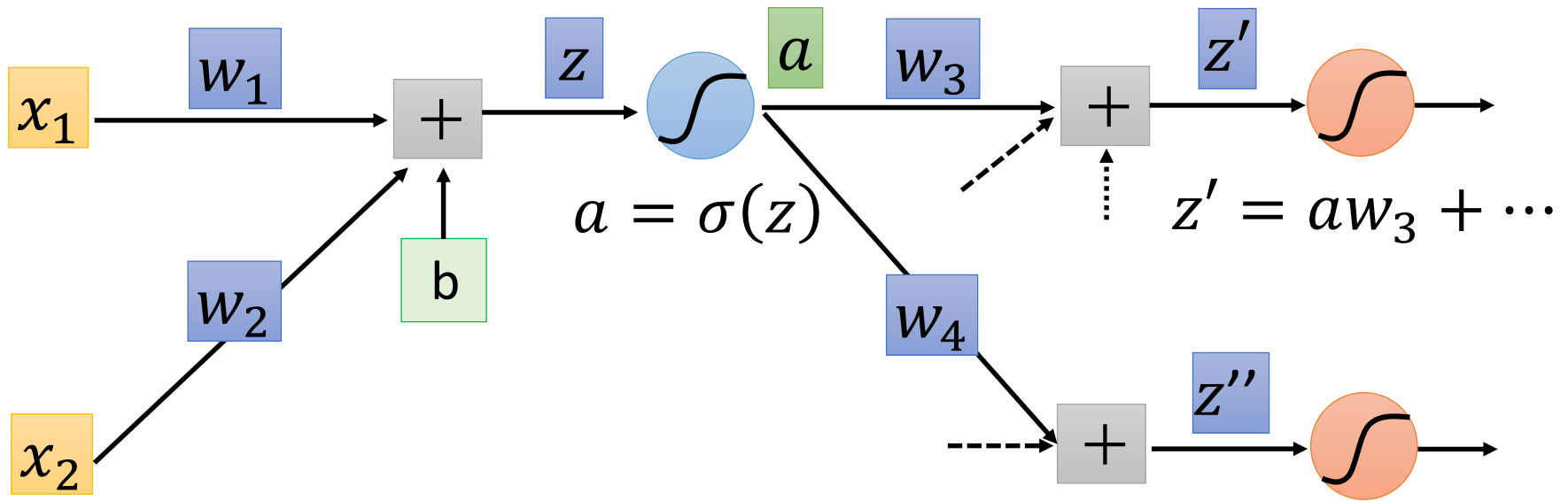
$$\frac{\partial l}{\partial z} = \frac{\partial a}{\partial z} \frac{\partial l}{\partial a}$$

➡ $\sigma'(z)$



Backpropagation – Backward pass

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



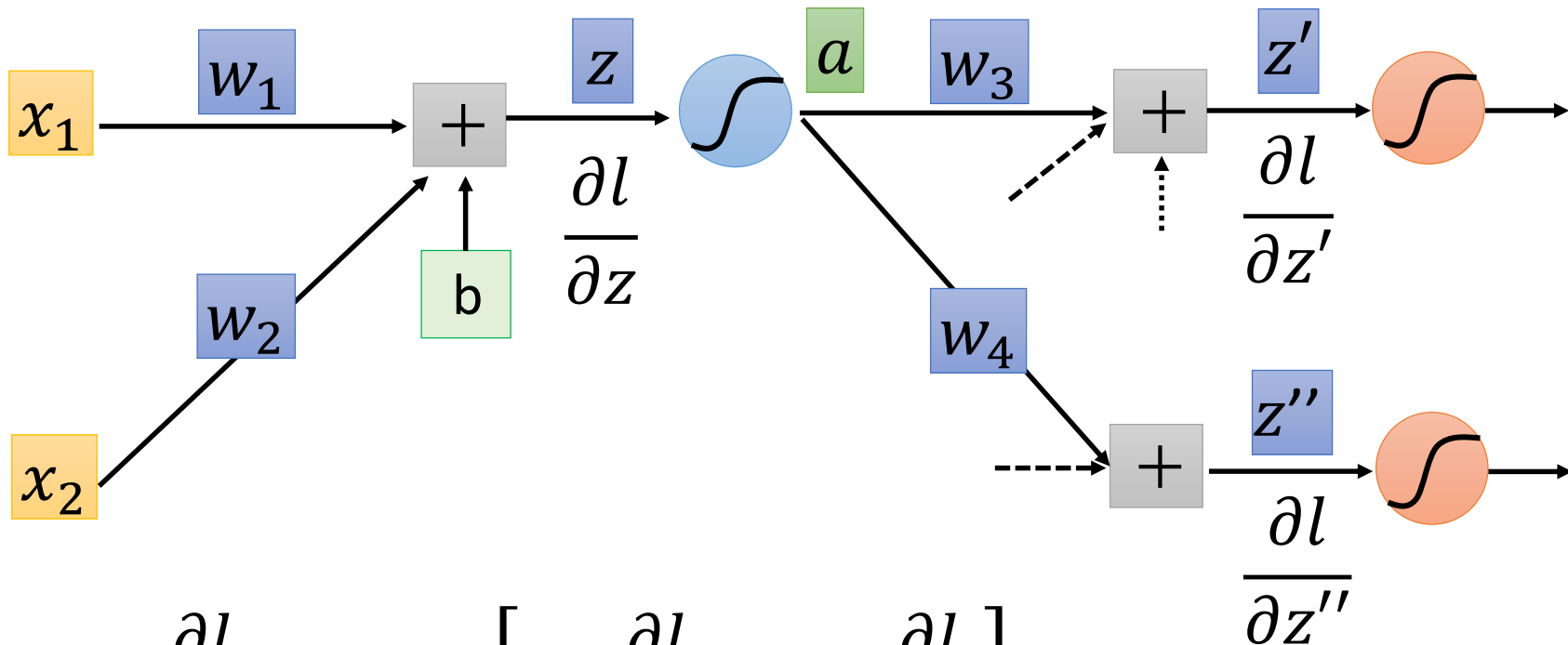
$$\frac{\partial l}{\partial z} = \frac{\partial a}{\partial z} \frac{\partial l}{\partial a}$$

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

Assumed
it's known

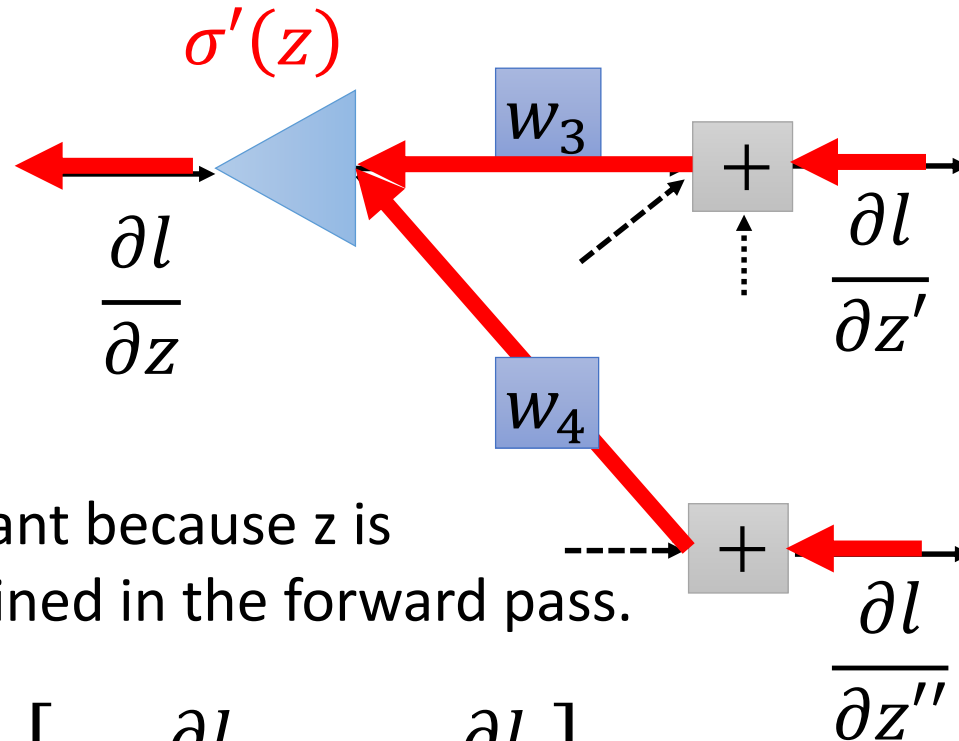
Backpropagation – Backward pass

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



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

Backpropagation – Backward pass

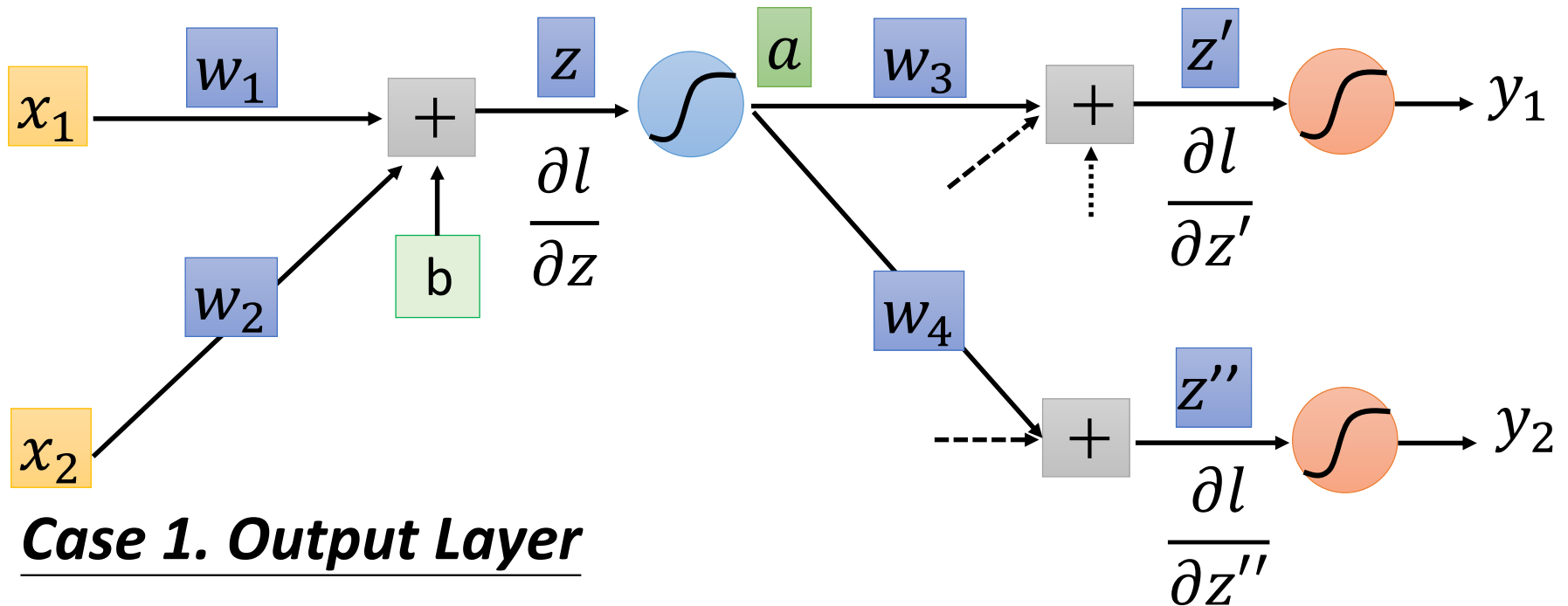


$\sigma'(z)$ is a constant because z is already determined in the forward pass.

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

Backpropagation – Backward pass

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



Case 1. Output Layer

$$\frac{\partial l}{\partial z'} = \frac{\partial y_1}{\partial z'} \frac{\partial l}{\partial y_1}$$

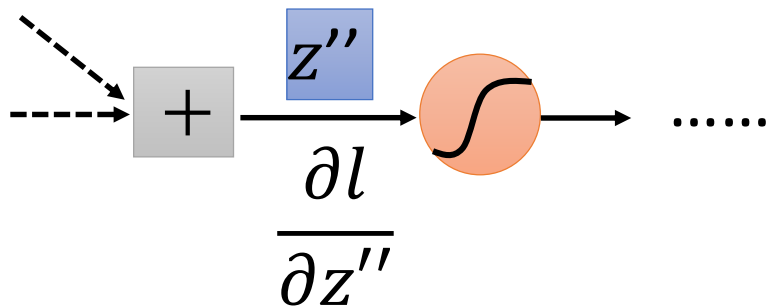
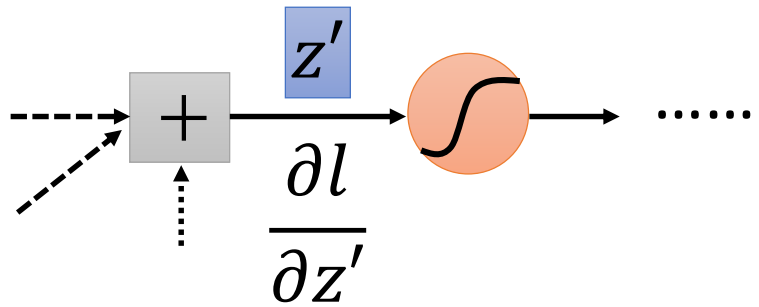
$$\frac{\partial l}{\partial z''} = \frac{\partial y_2}{\partial z''} \frac{\partial l}{\partial y_2}$$

Done!

Backpropagation – Backward pass

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

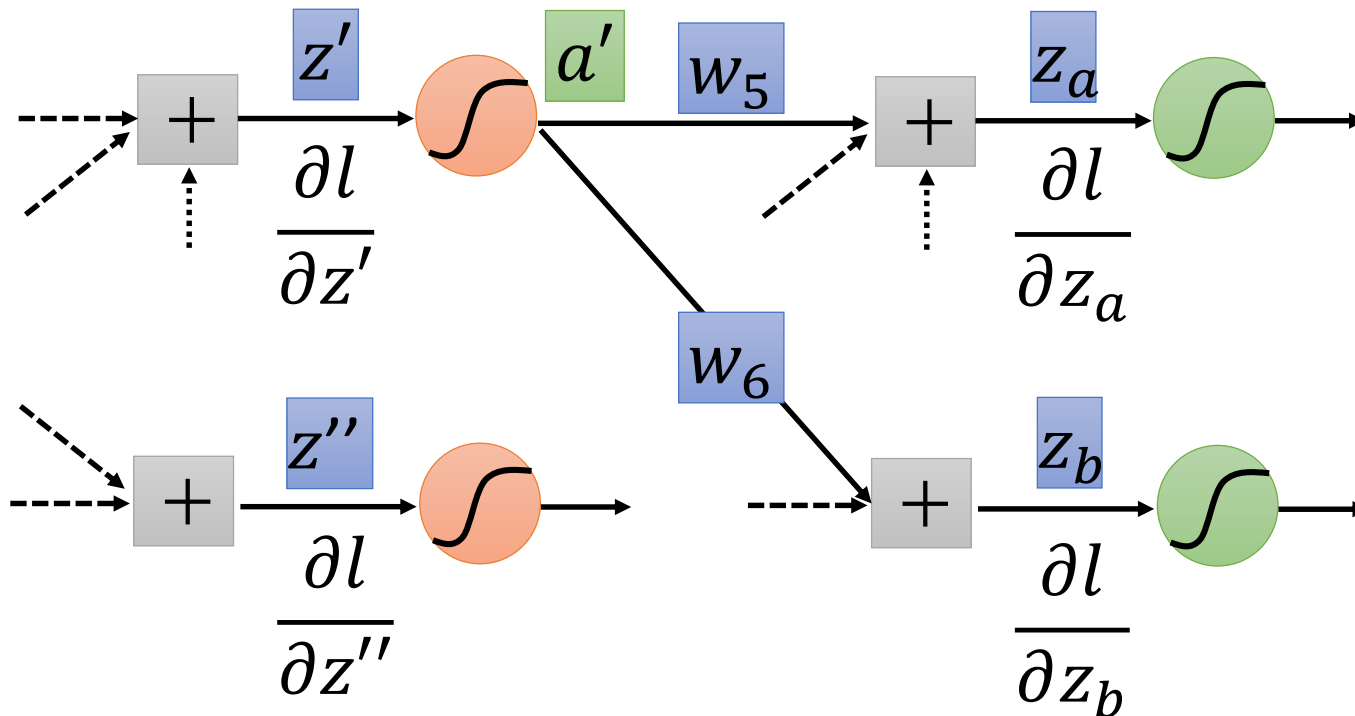
Case 2. Not Output Layer



Backpropagation – Backward pass

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

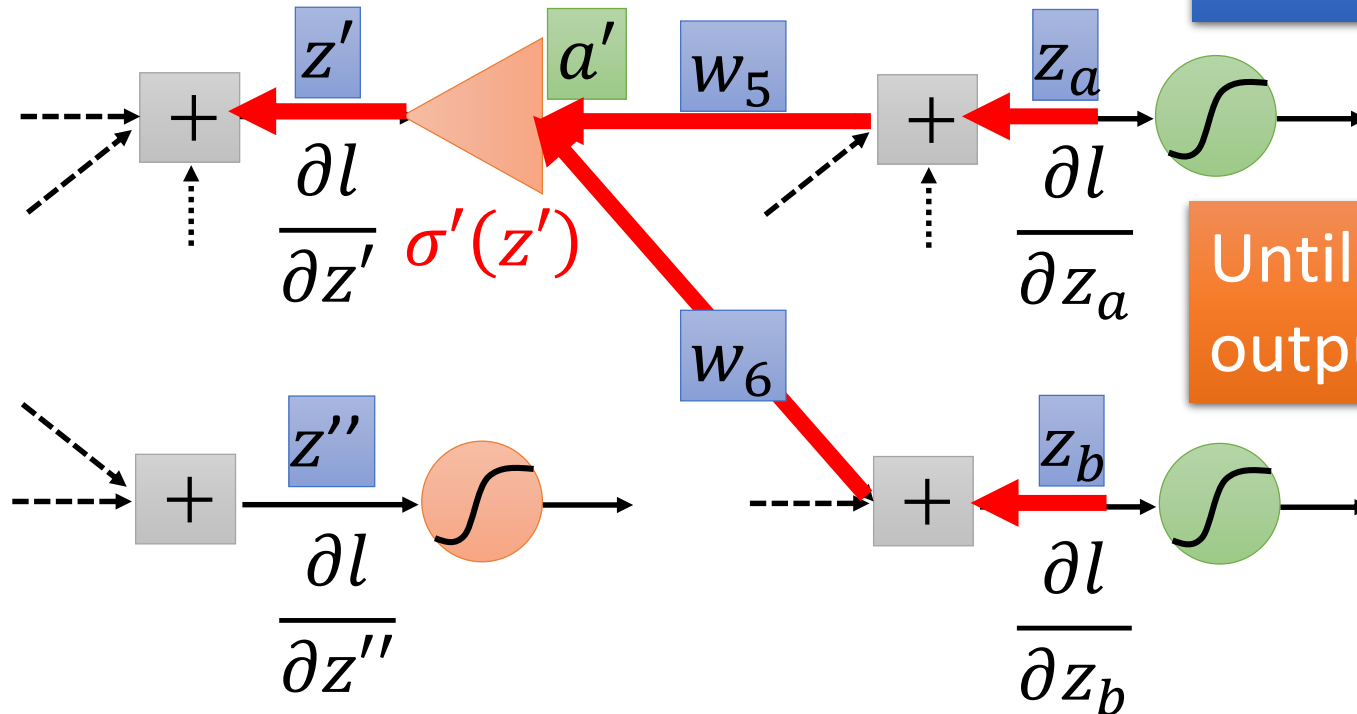
Case 2. Not Output Layer



Backpropagation – Backward pass

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

Case 2. Not Output Layer



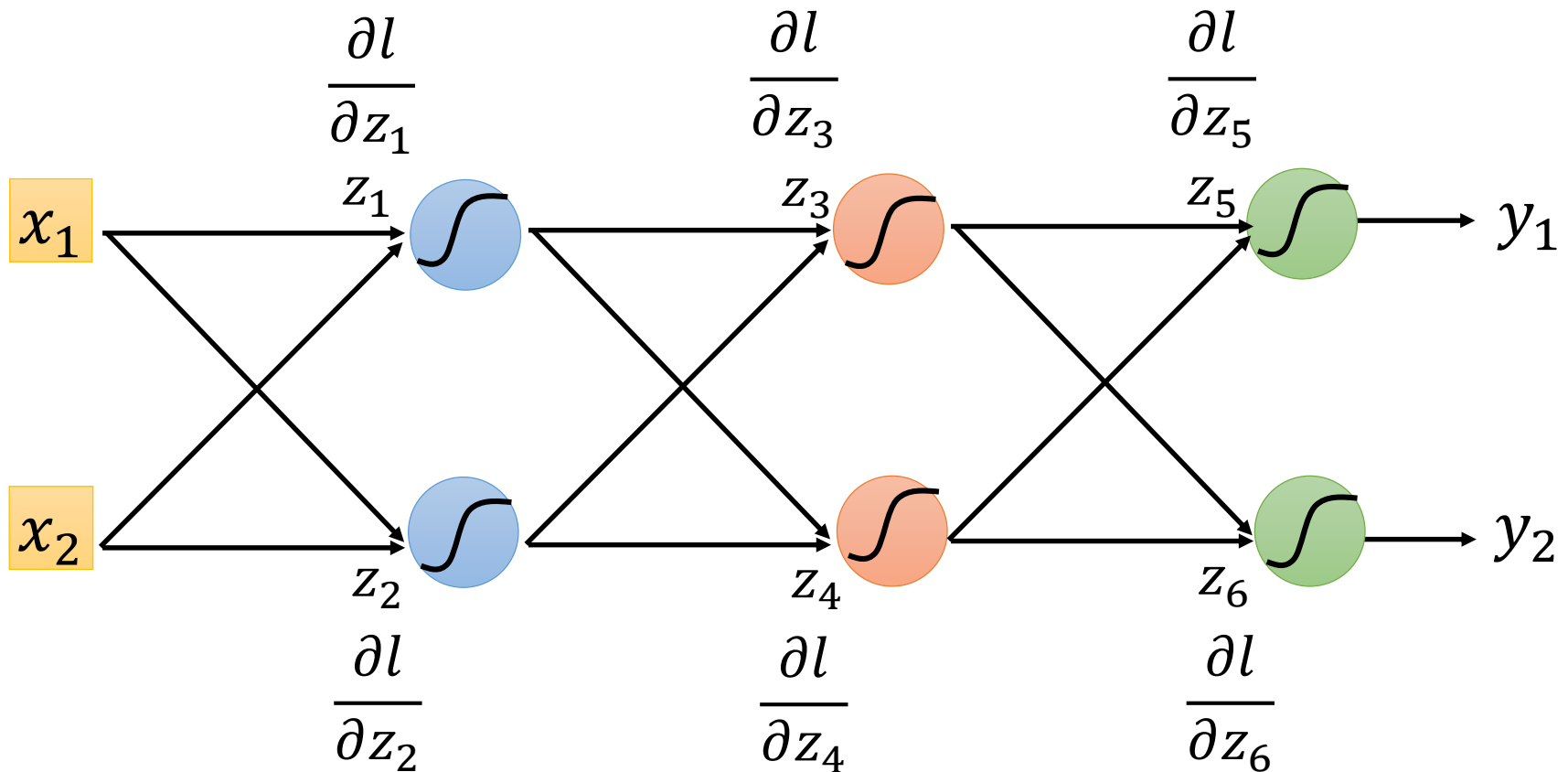
Compute $\partial l / \partial z$
recursively

Until we reach the
output layer

Backpropagation – Backward Pass

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

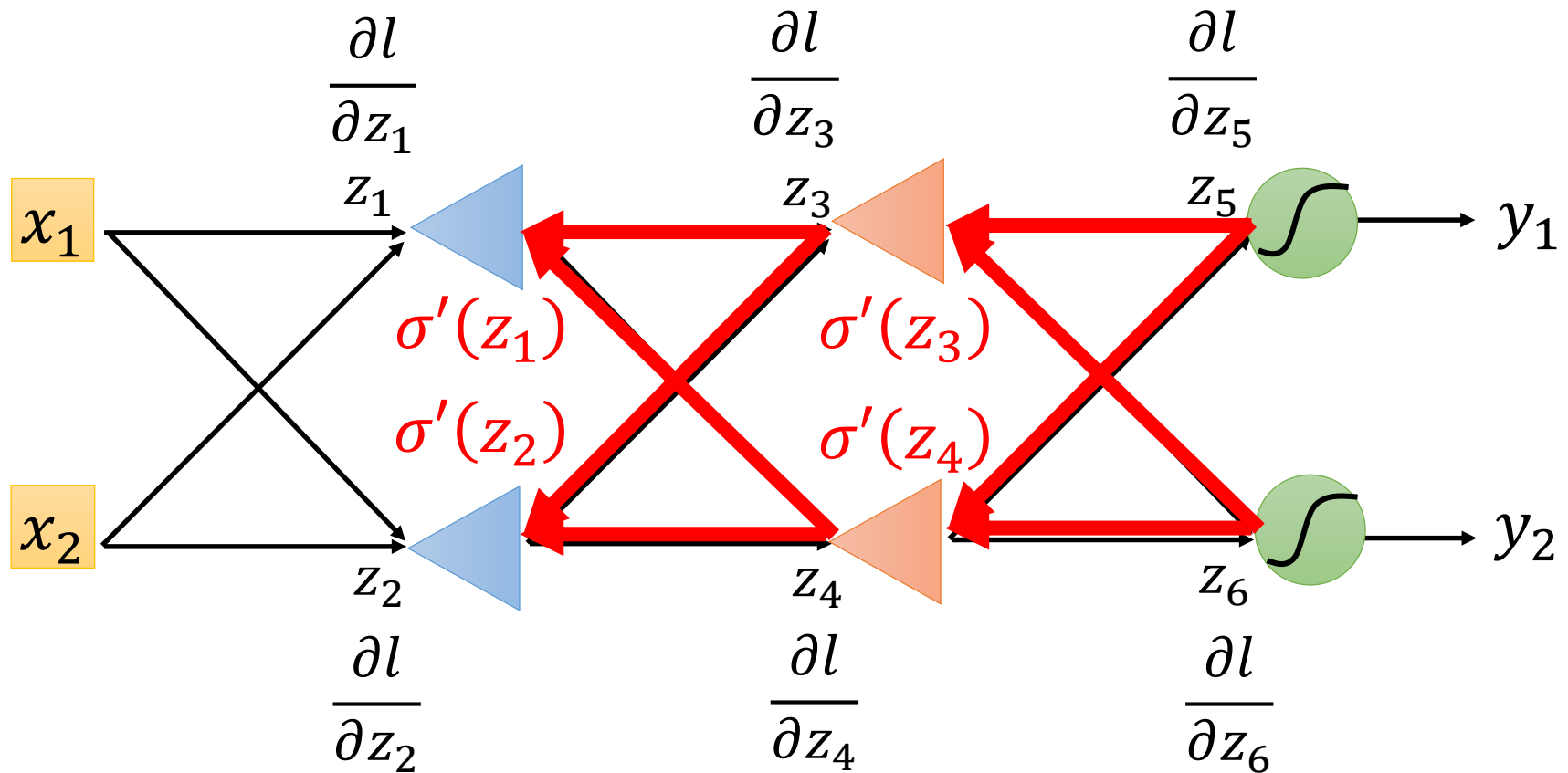
Compute $\partial l / \partial z$ from the output layer



Backpropagation – Backward Pass

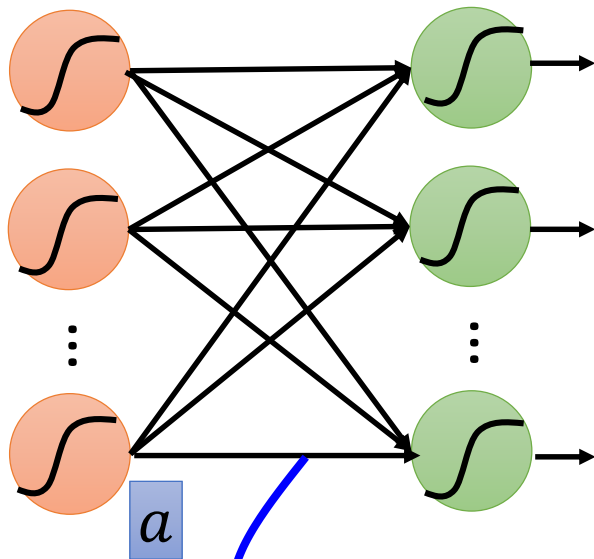
Compute $\partial l / \partial z$ for all activation function inputs z

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



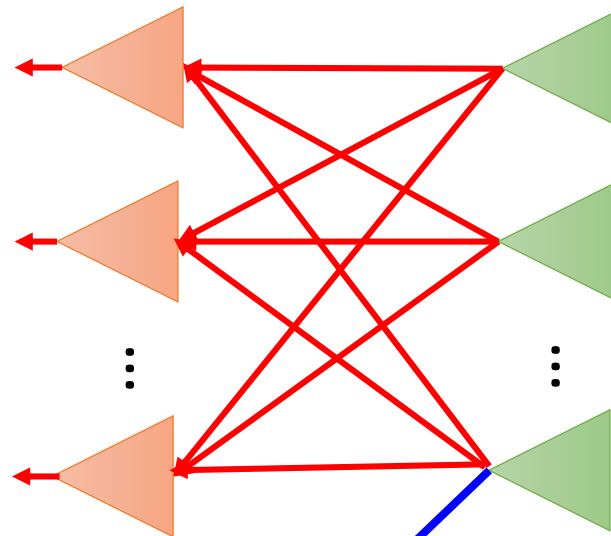
Backpropagation – Summary

Forward Pass



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

Backward Pass



\times

$$\frac{\partial l}{\partial z}$$

$$= \frac{\partial l}{\partial w}$$

for all w