Backpropagation

Hung-yi Lee
李宏毅

Gradient Descent

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

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

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

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

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

Millions of parameters ......

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

Chain Rule

Case 1

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

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

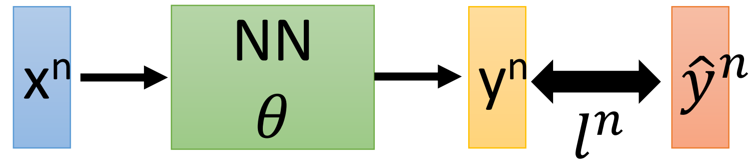
Case 2

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

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

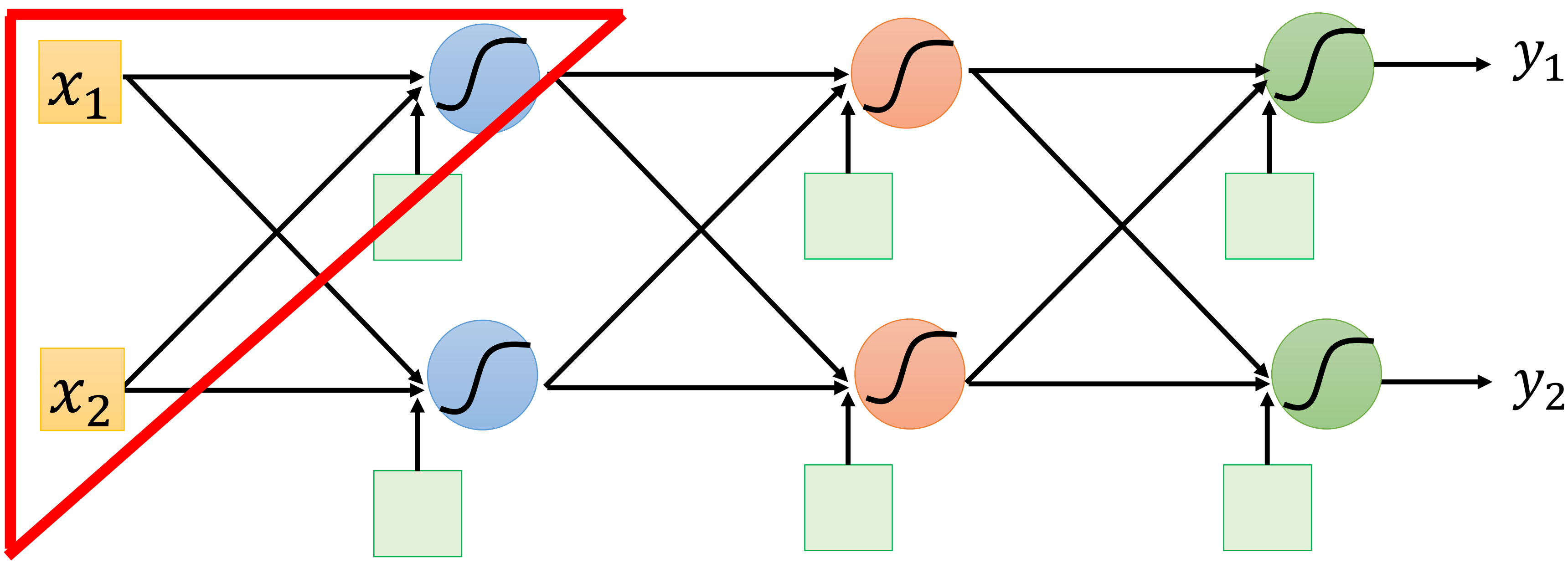
$$\frac{dz}{ds} = \frac{\partial z}{\partial x}\frac{dx}{ds} + \frac{\partial z}{\partial y}\frac{dy}{ds}$$

Backpropagation

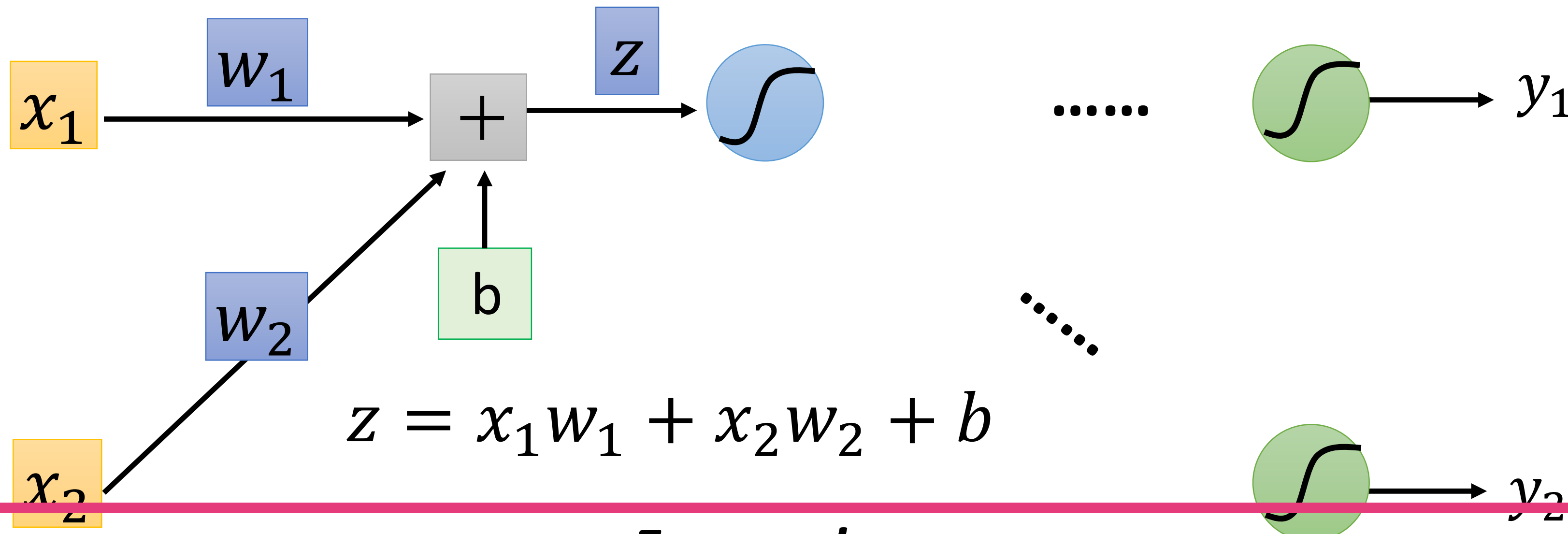


每一筆data的loss的sum

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



Backpropagation



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

Forward pass:

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

Backward pass:

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

某一比data的loss

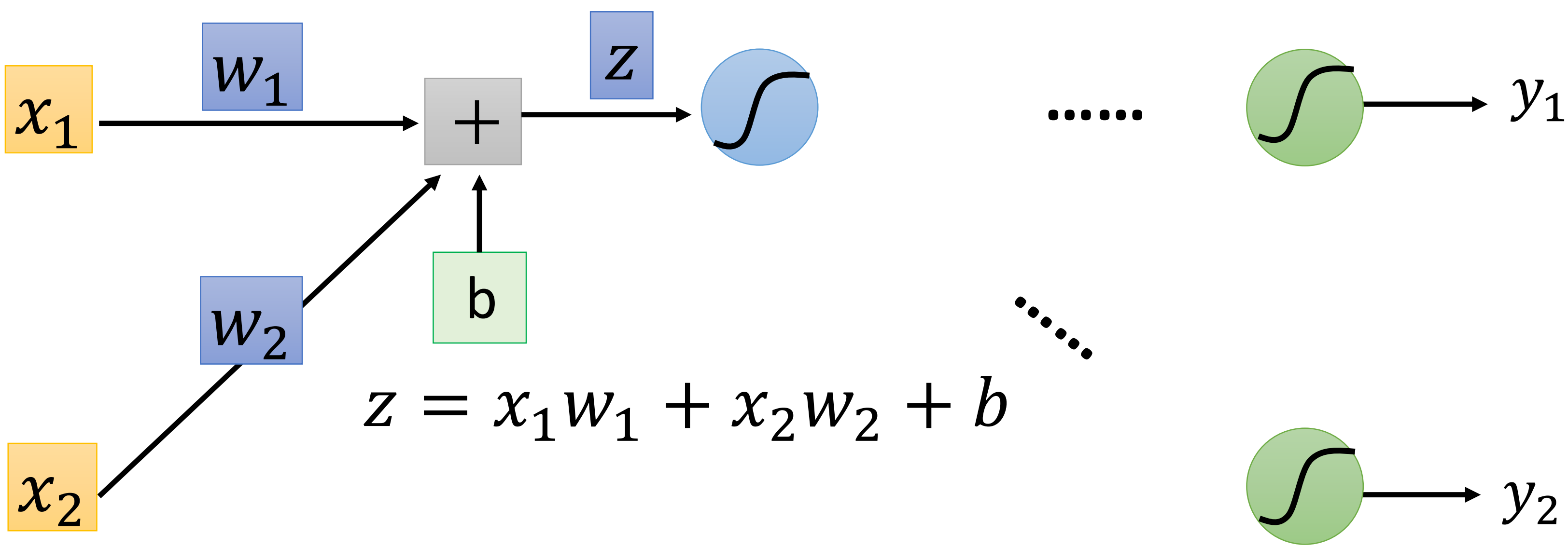
$$\frac{\partial l}{\partial w} = ? \quad \frac{\partial z}{\partial w} \frac{\partial l}{\partial z}$$

(Chain rule)

算這兩個偏微分，一個用forward一個用backward

Backpropagation – Forward pass

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



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

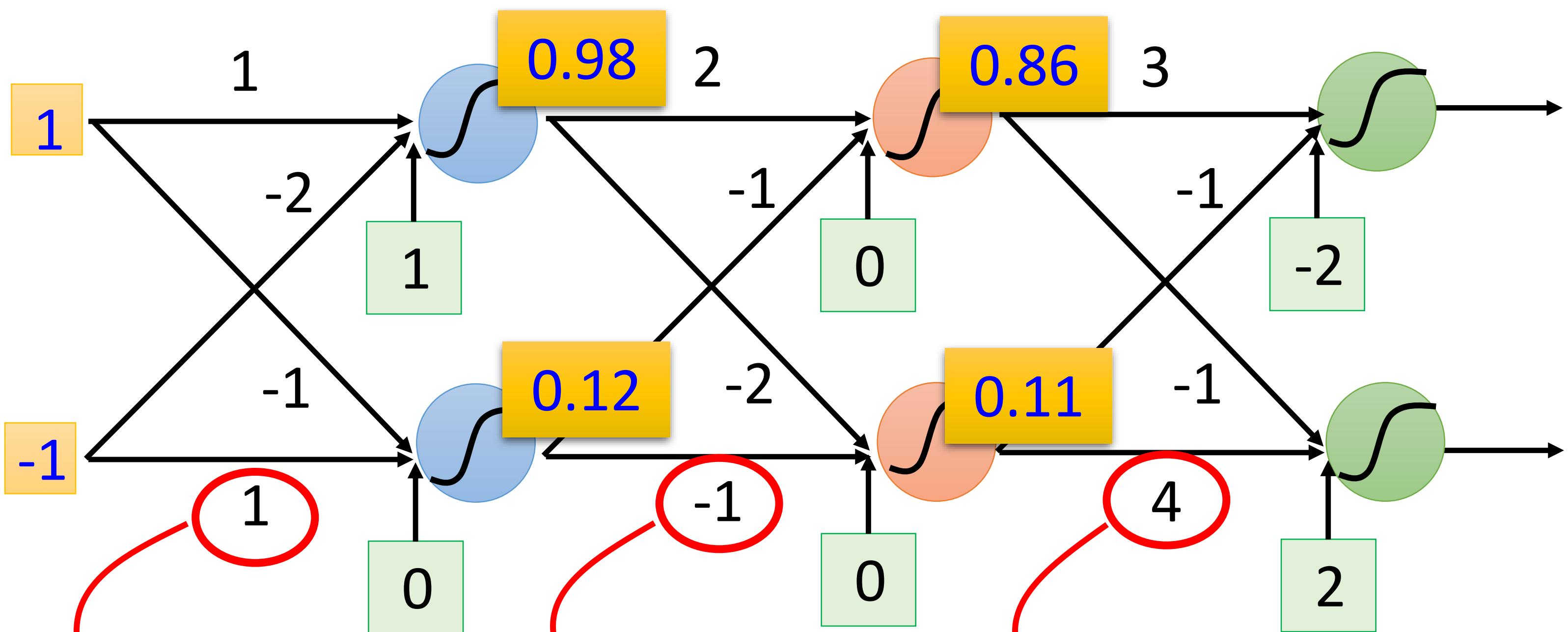
$$\left.\begin{aligned} \partial z/\partial w_1 &=?\ \ x_1 \\ \partial z/\partial w_2 &=?\ \ x_2 \end{aligned}\right\}$$

The value of the input connected by the weight

Backpropagation – Forward pass

對z做偏微分

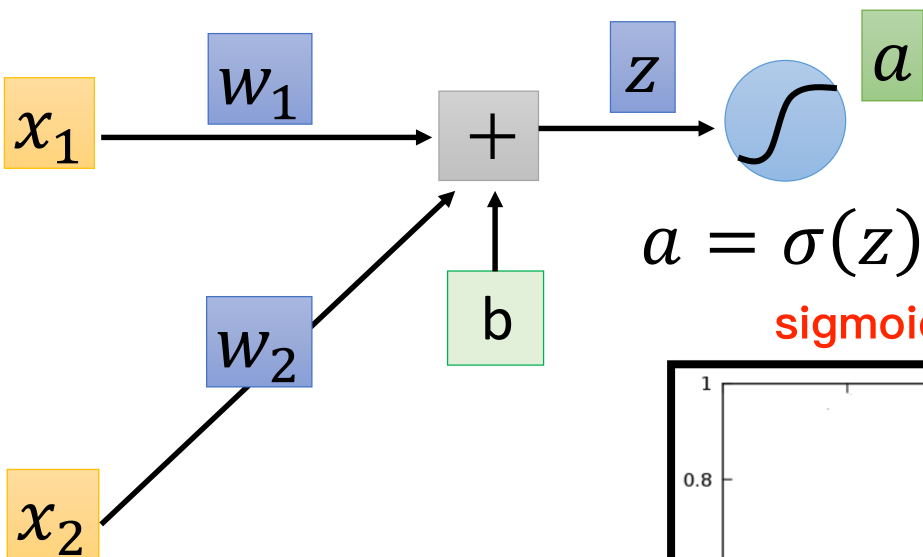
Compute $\partial z/\partial w$ for all parameters



$$\frac{\partial z}{\partial w} = -1 \qquad \frac{\partial z}{\partial w} = 0.12 \qquad \frac{\partial z}{\partial w} = 0.11$$

Backpropagation – Backward pass

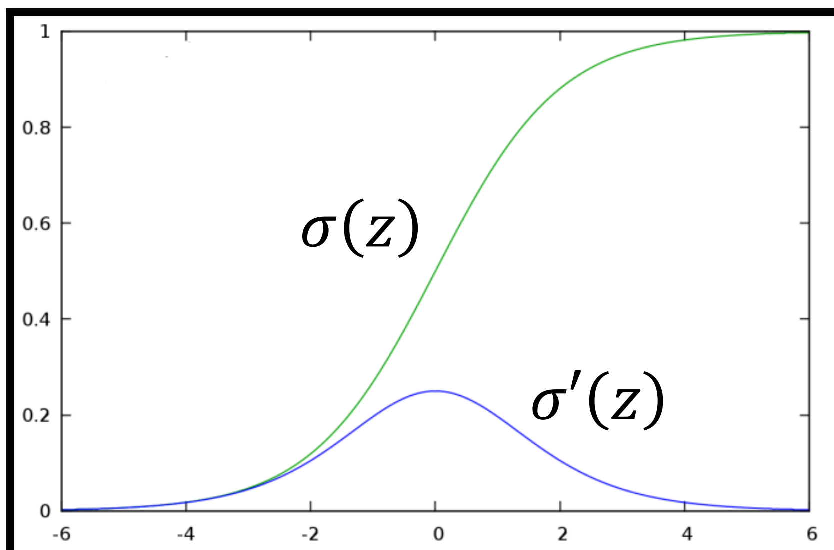
Compute $\partial l / \partial z$ for all activation function inputs z



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

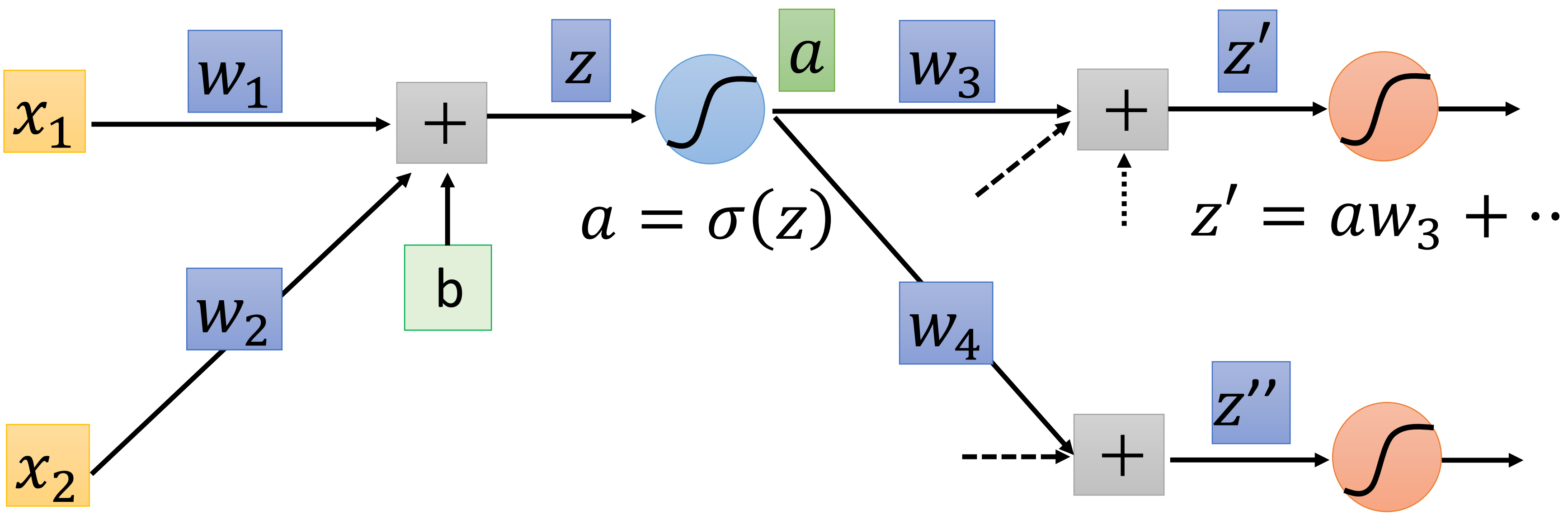
$$\Rightarrow \sigma'(z)$$

sigmoid function做微分



Backpropagation – Backward pass

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



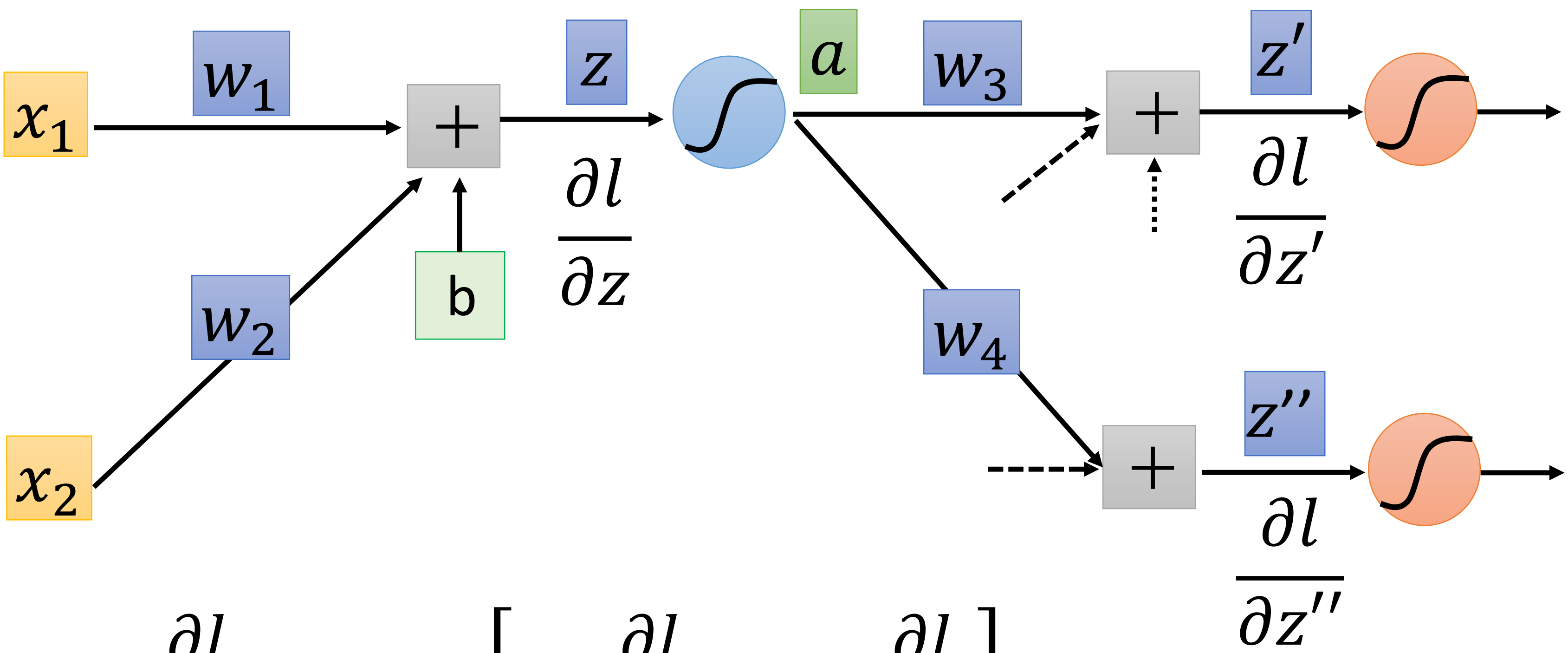
$$\frac{\partial l}{\partial z} = \frac{\partial a}{\partial z}\frac{\partial l}{\partial a} \qquad \frac{\partial l}{\partial a} = \underbrace{\frac{\partial z'}{\partial a}}_{w_3}\underbrace{\frac{\partial l}{\partial z'}}_{?} + \underbrace{\frac{\partial z''}{\partial a}}_{w_4}\underbrace{\frac{\partial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

因為a會影響兩個參數

Assumed it's known

Backpropagation – Backward pass

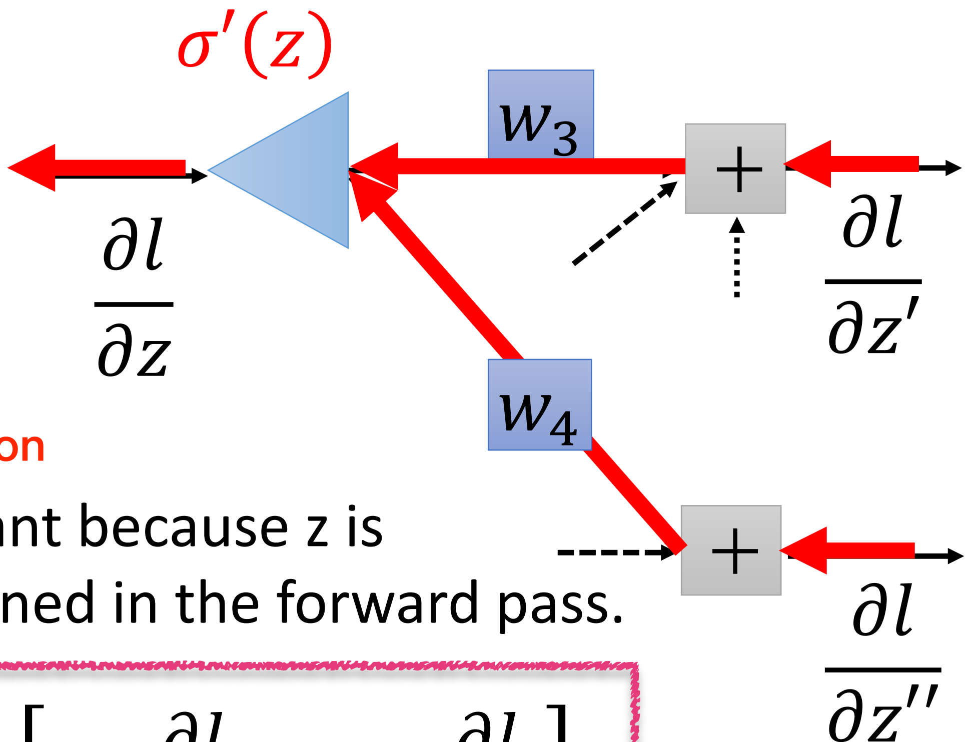
Compute $\partial l/\partial z$ for all activation function inputs z



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

看作是另外一個linear的neural

Backpropagation – Backward pass



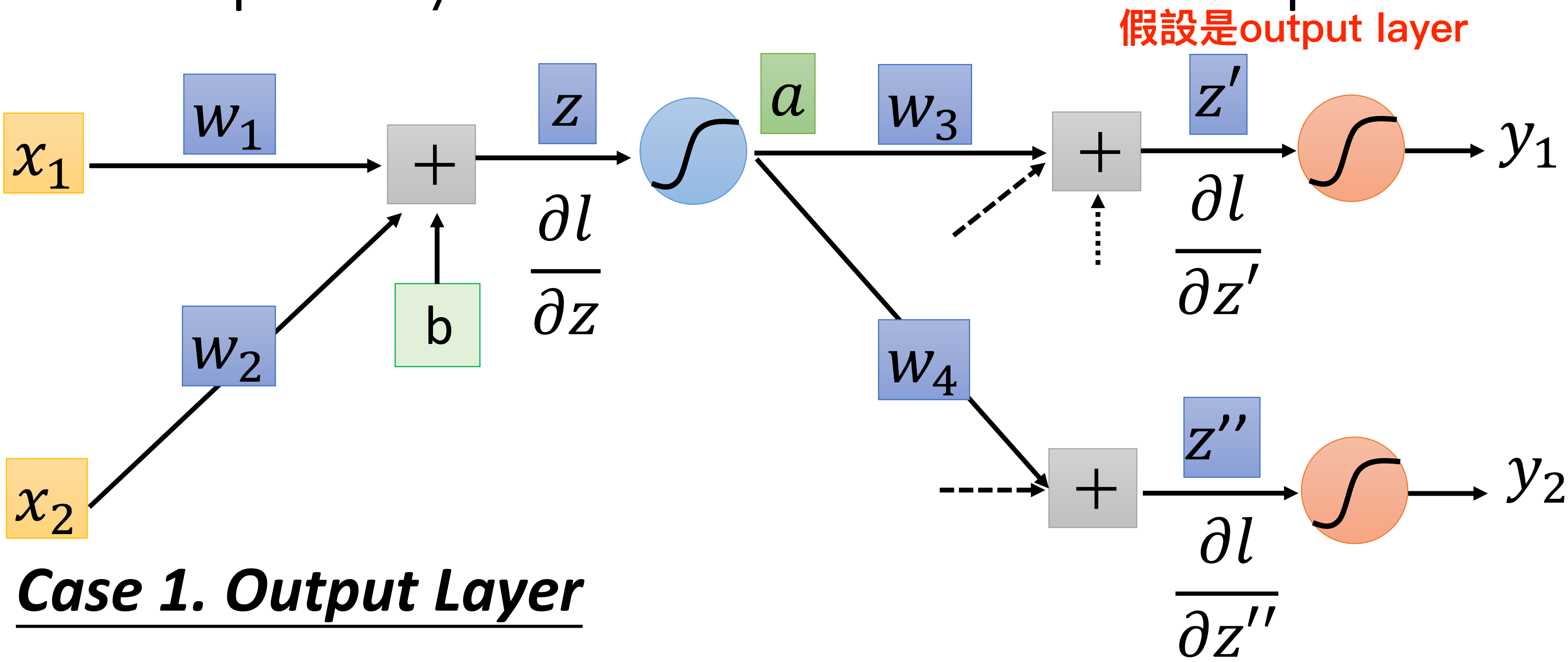
$\sigma'()$ = activation function

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

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

Backpropagation – Backward pass

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



Case 1. Output Layer

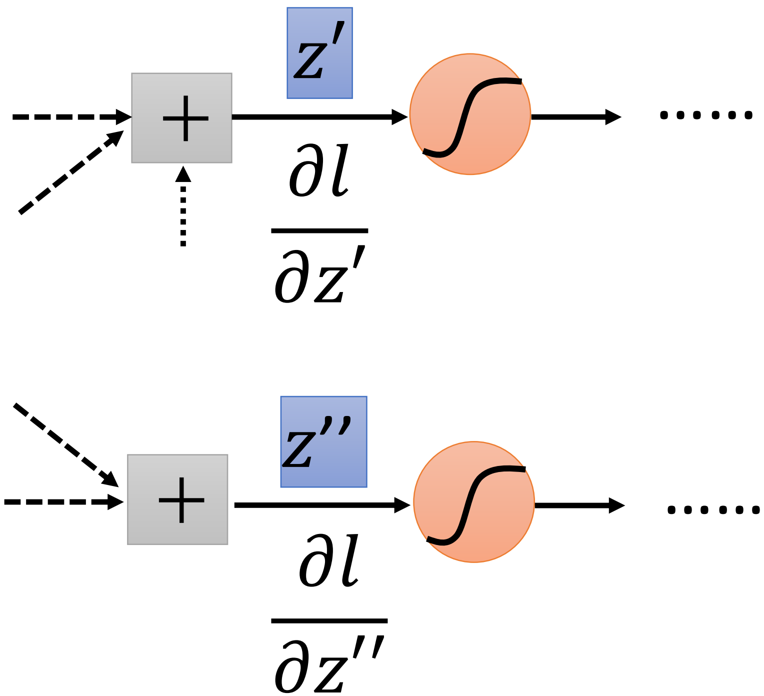
$$\frac{\partial l}{\partial z'} = \frac{\partial y_1}{\partial z'} \frac{\partial l}{\partial y_1} \qquad \frac{\partial l}{\partial z''} = \frac{\partial y_2}{\partial z''} \frac{\partial l}{\partial y_2}$$

Done!

Backpropagation – Backward pass

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

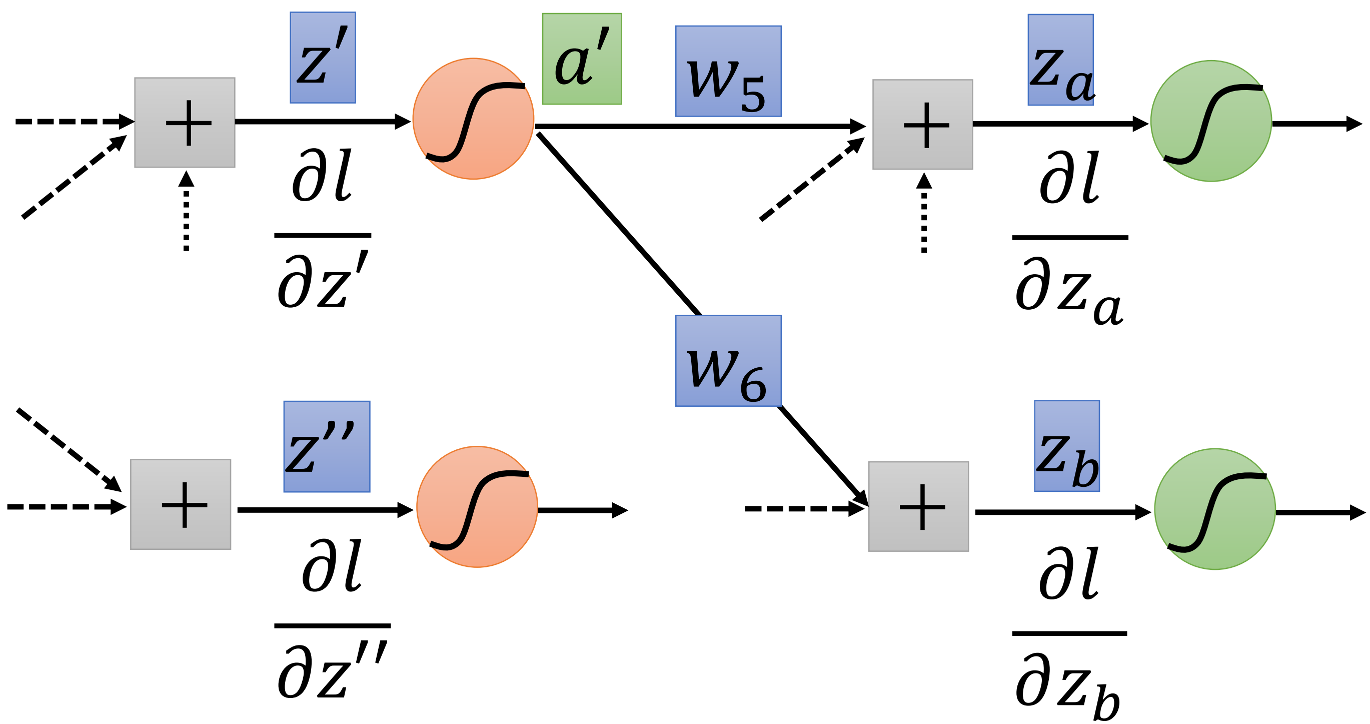
Case 2. Not Output Layer



Backpropagation – Backward pass

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

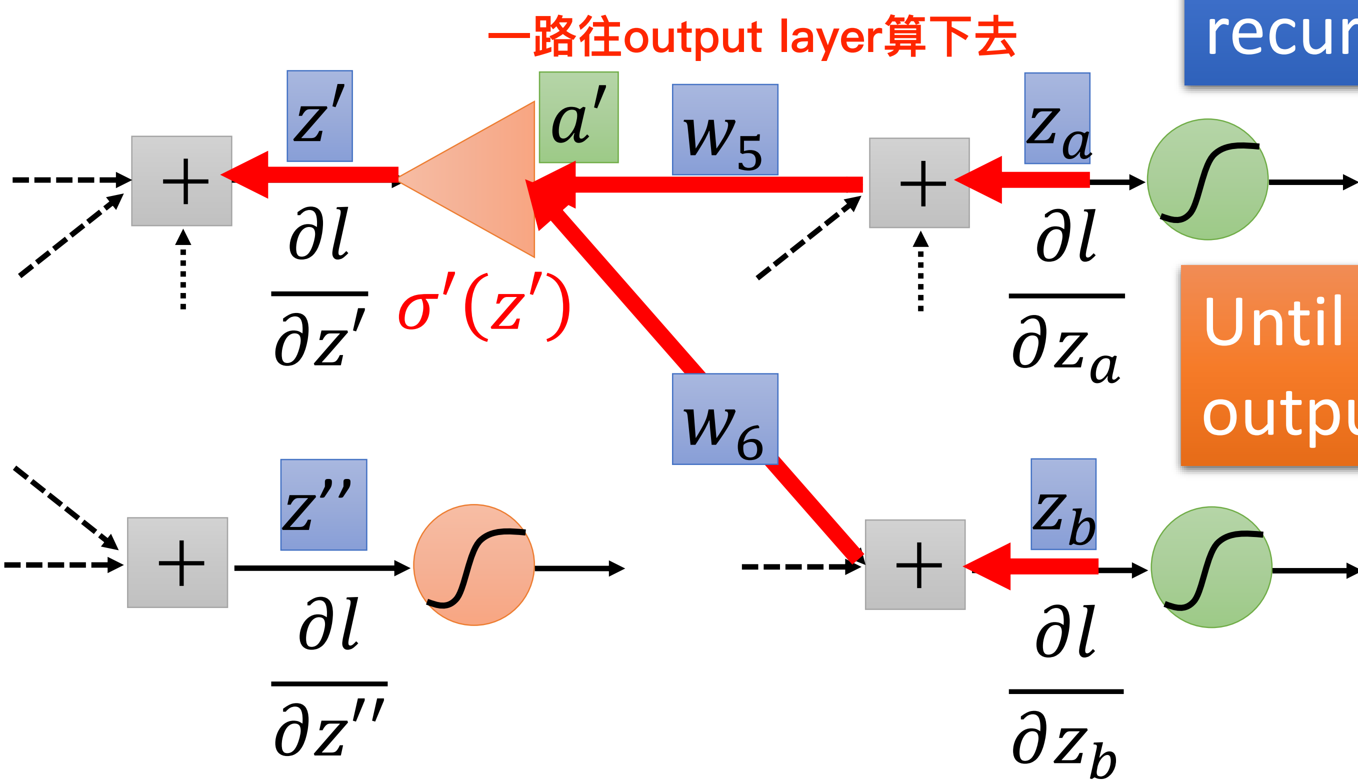
Case 2. Not Output Layer



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

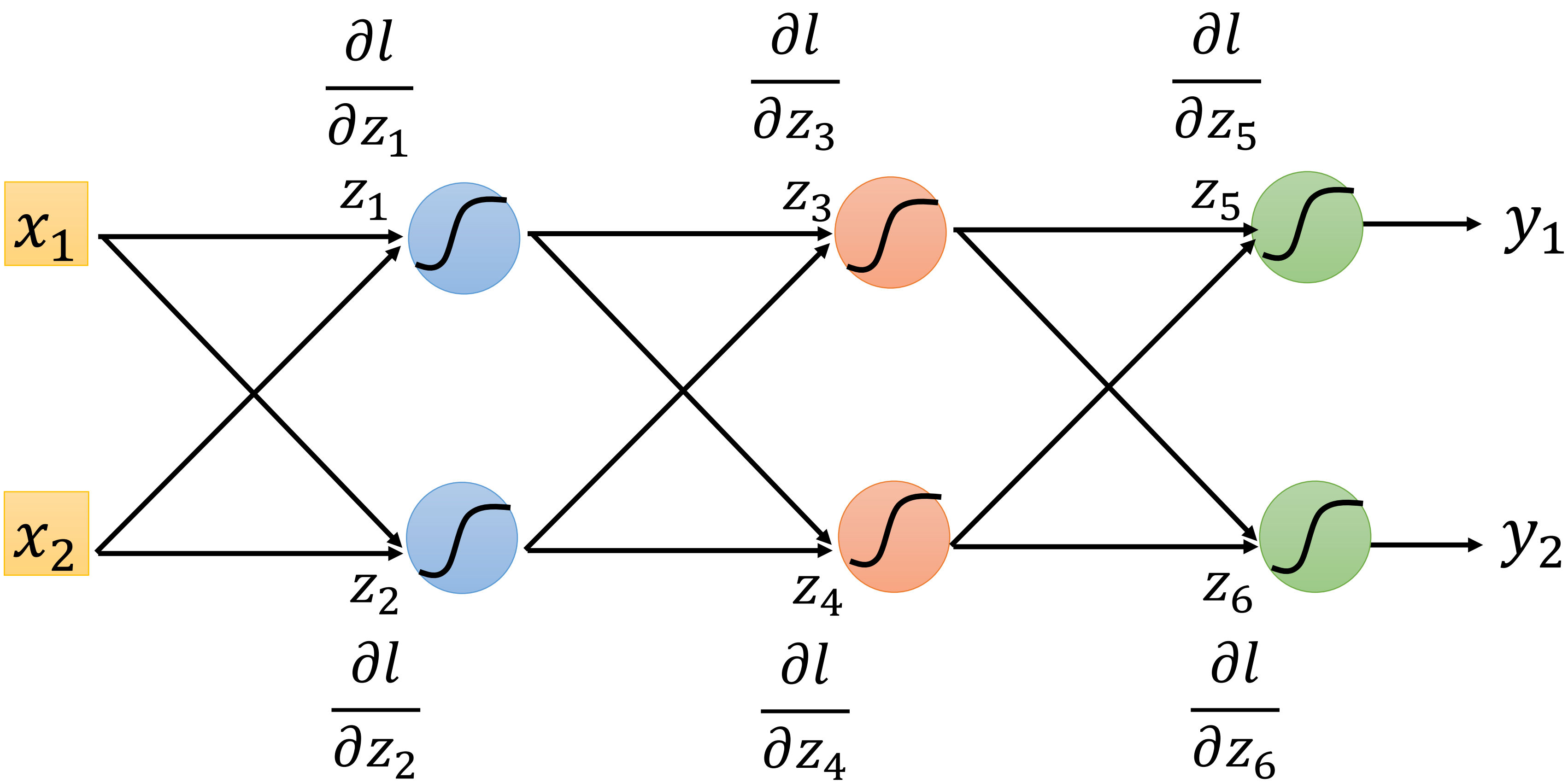
Until we reach the output layer ……

Backpropagation – Backward Pass

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

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

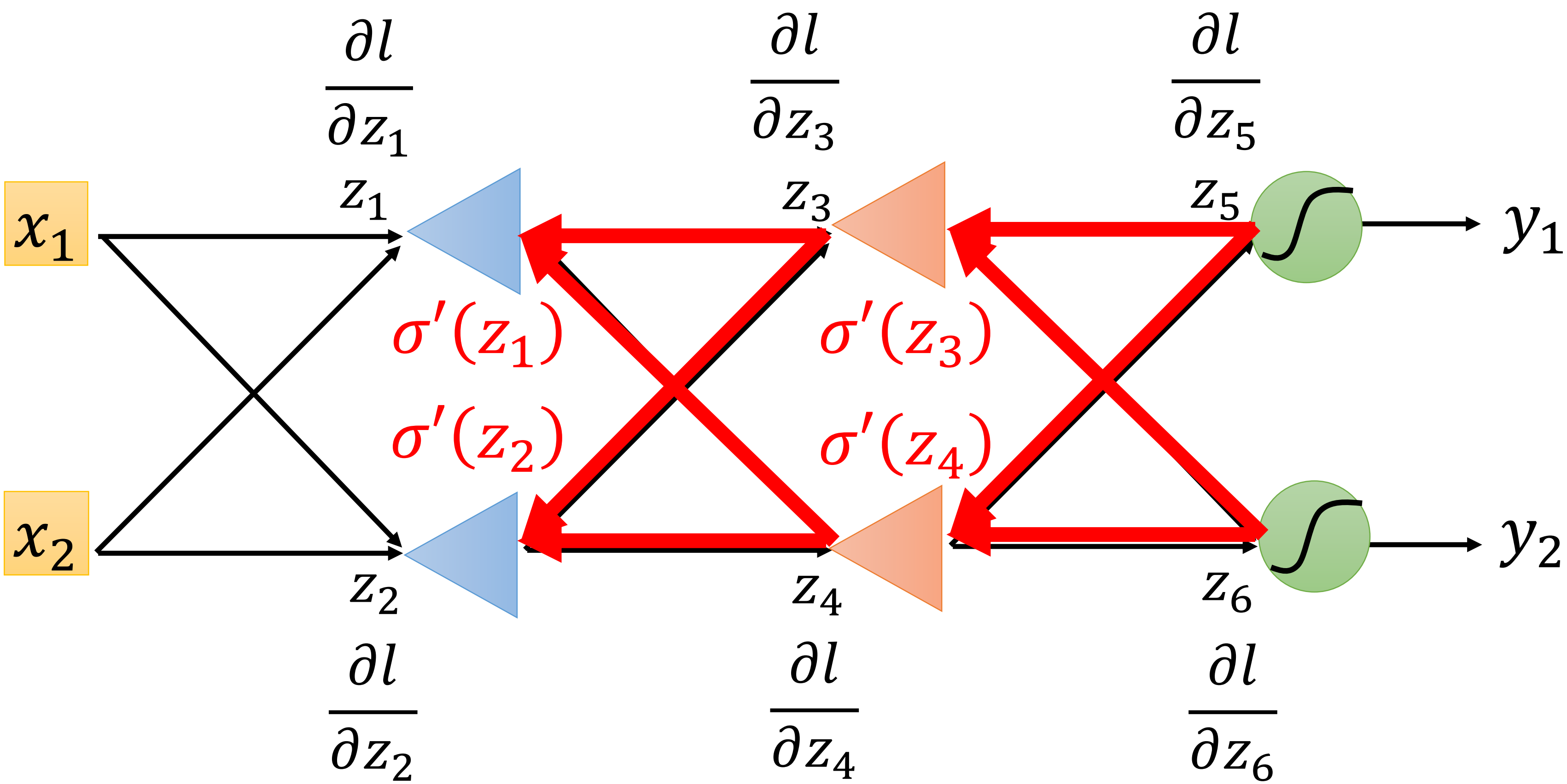
如果從z5,z6開始算
會比較簡單



Backpropagation – Backward Pass

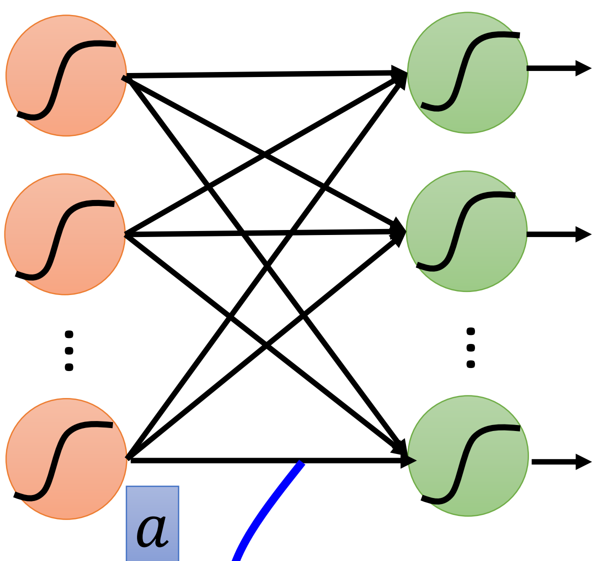
Compute $\partial l/\partial z$ for all activation function inputs z

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

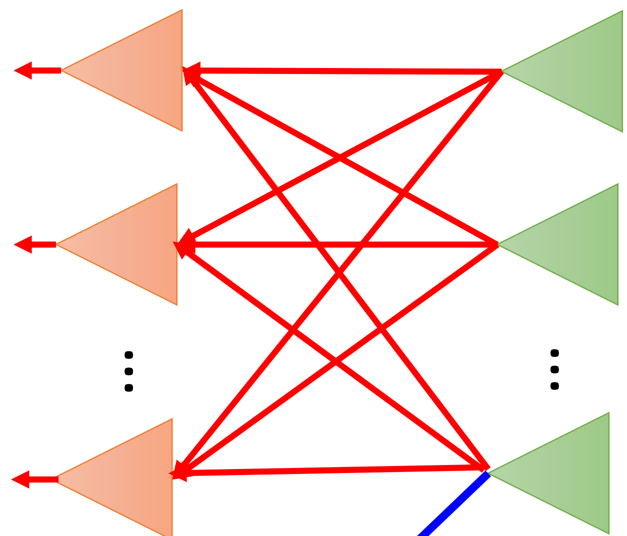


Backpropagation – Summary

Forward Pass



Backward Pass



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

找到gradient後就可以做gradient descend了

for all w