Backpropagation

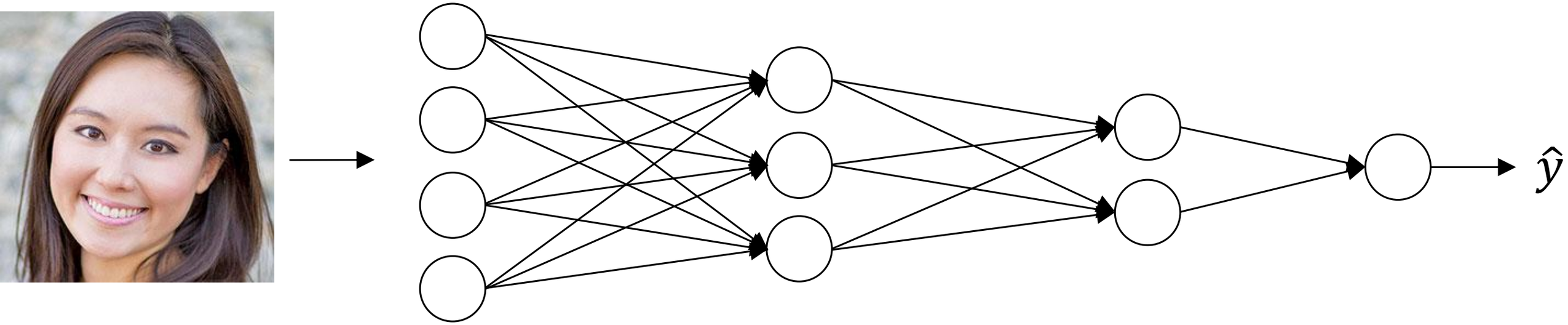
Seyoung Yun

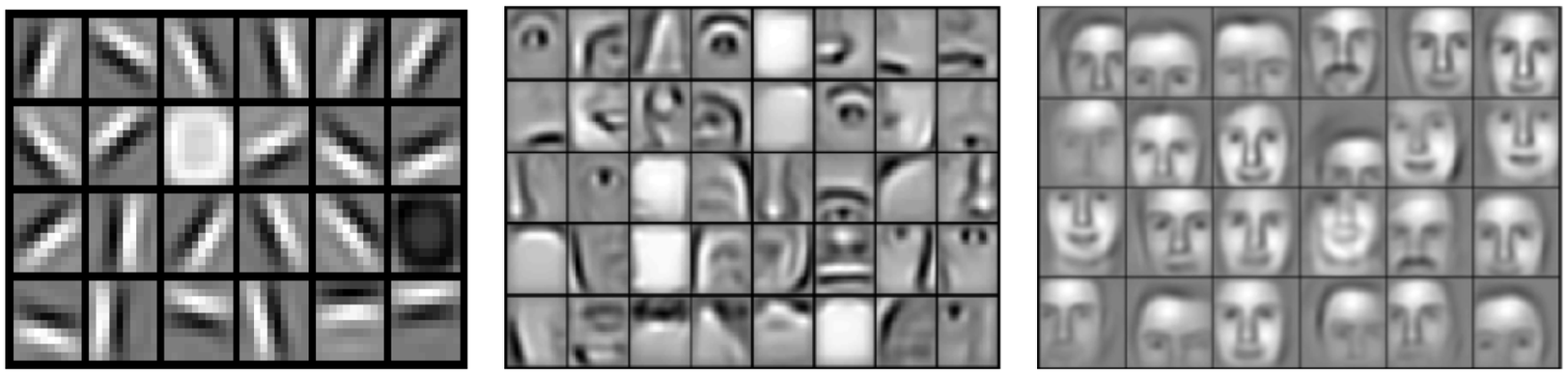
- http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture4.pdf

- http://cs231n.github.io/optimization-2/

- http://yann.lecun.com/exdb/publis/pdf/lecun-98b.pdf

Intuition about deep representation

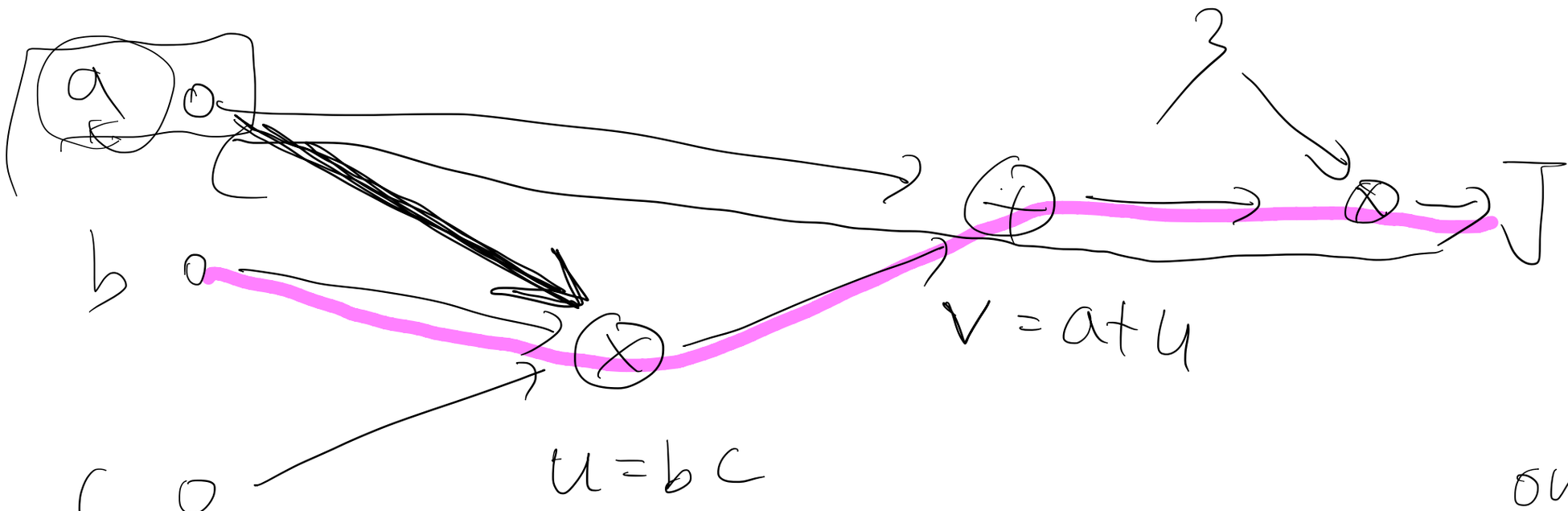




Informally: There are functions you can compute with a “small” L-layer deep neural network that shallower networks require exponentially more hidden units to compute.

Computational Graph

$J(a,b,c) = 3(a+bc)$



input

$$\frac{\partial J}{\partial a} = \frac{\partial J}{\partial v}\cdot\frac{\partial v}{\partial a} = 3$$

$\frac{\partial J}{\partial v} = 3$

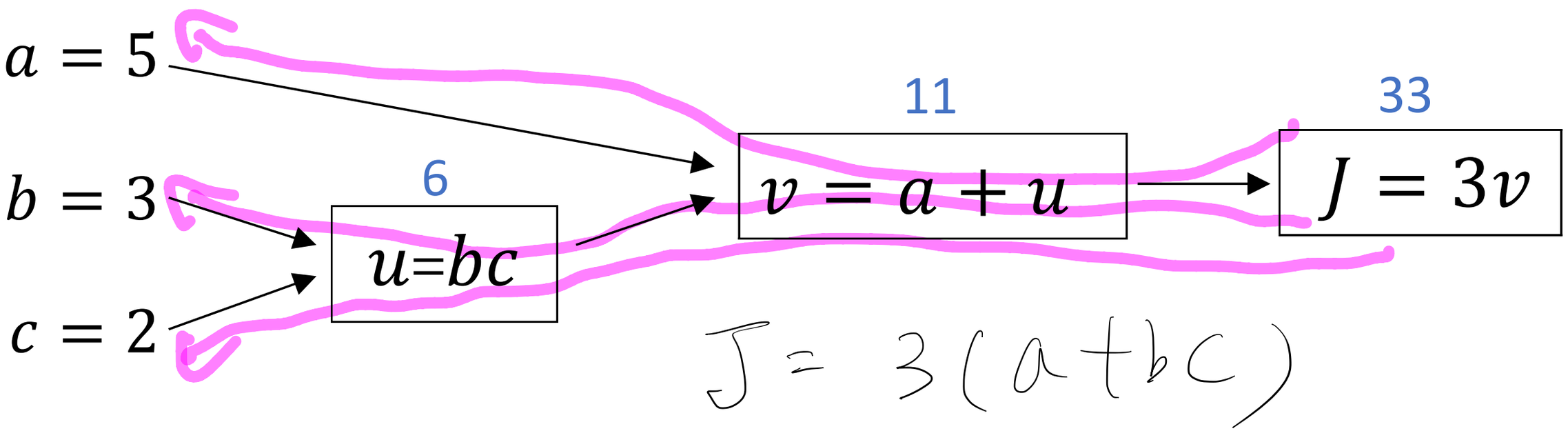
$\frac{\partial v}{\partial a} = 1$

$$\frac{\partial J}{\partial b} = \frac{\partial J}{\partial v}\cdot\frac{\partial v}{\partial u}\cdot\frac{\partial u}{\partial b}$$

$\frac{\partial J}{\partial v} = 3$ $\quad \frac{\partial v}{\partial u} = 1$ $\quad \frac{\partial u}{\partial b} = c$

$= 3c$

Computing derivatives

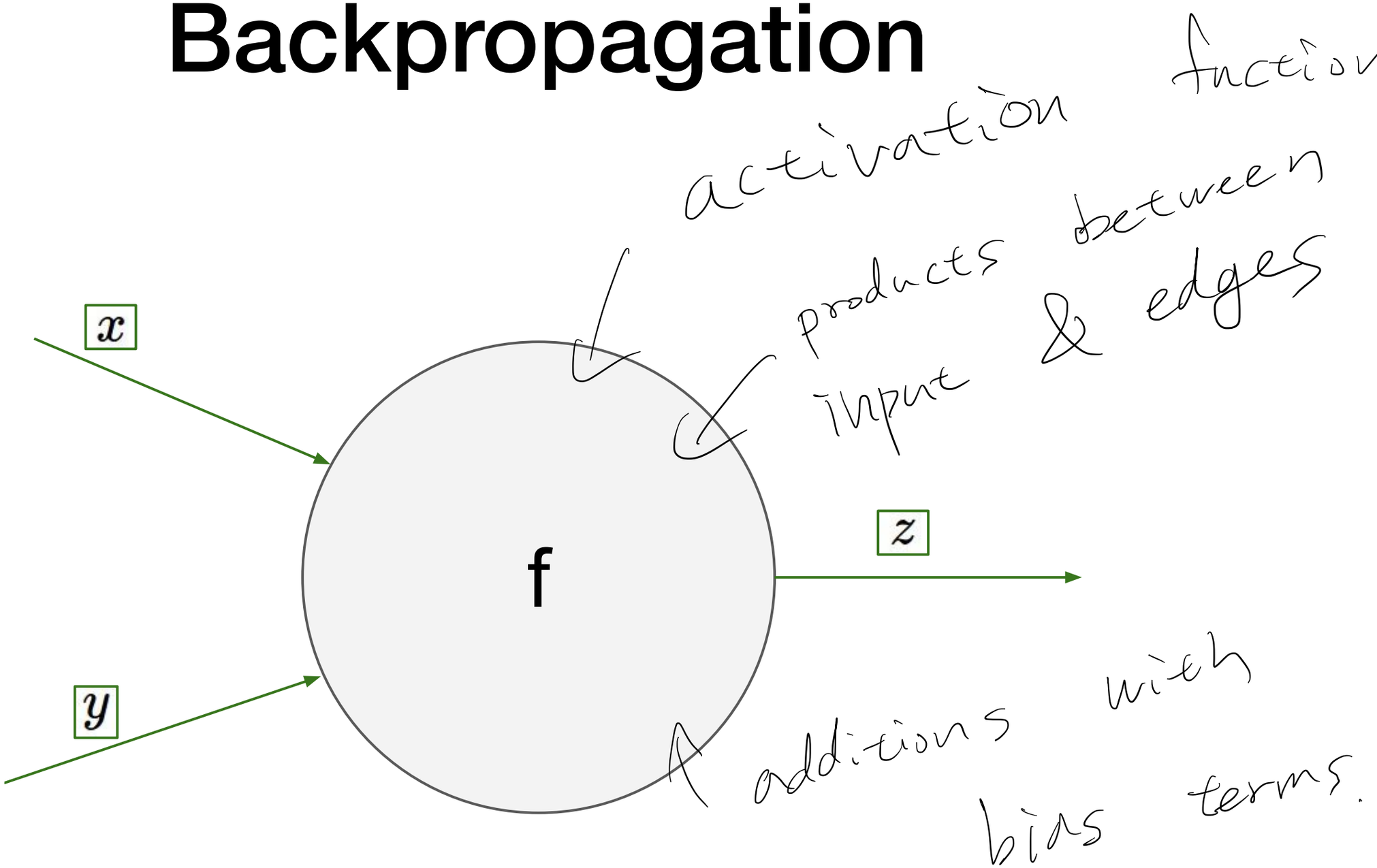


$$\frac{\partial J}{\partial a} =$$

$$\frac{\partial J}{\partial b} =$$

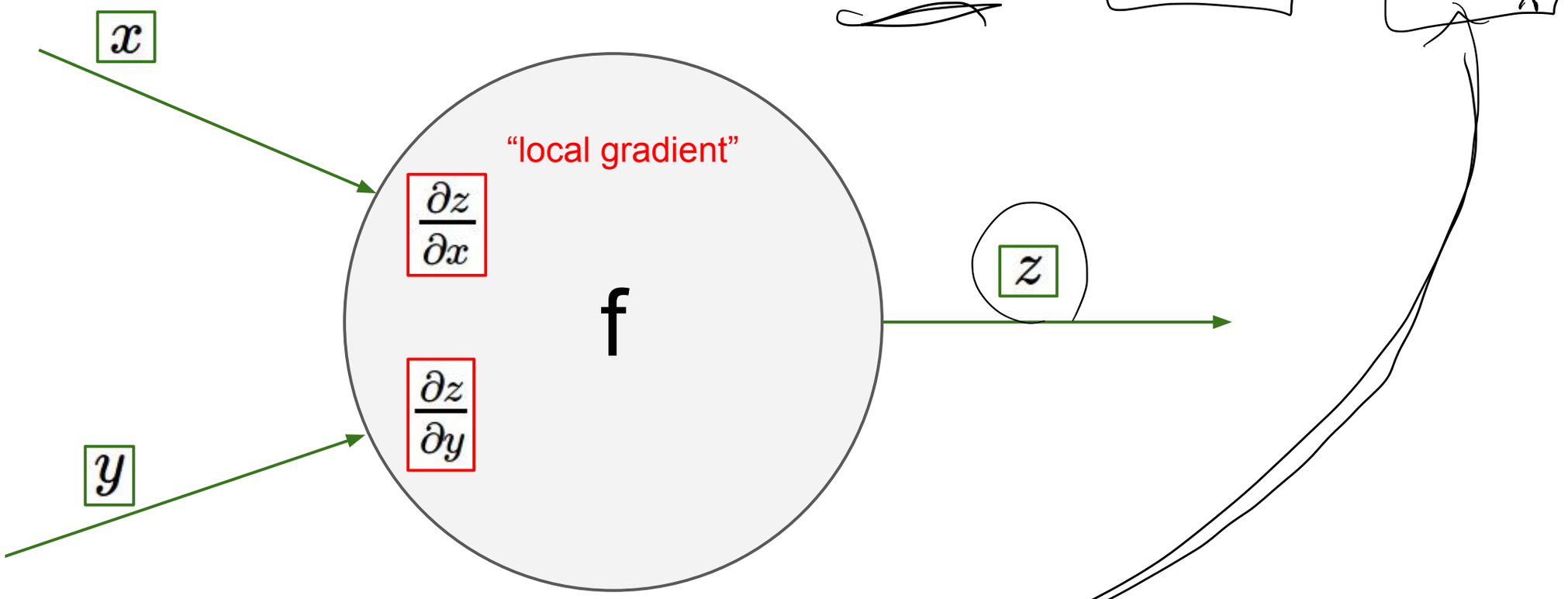
$$\frac{\partial J}{\partial c} =$$

Backpropagation



Backpropagation

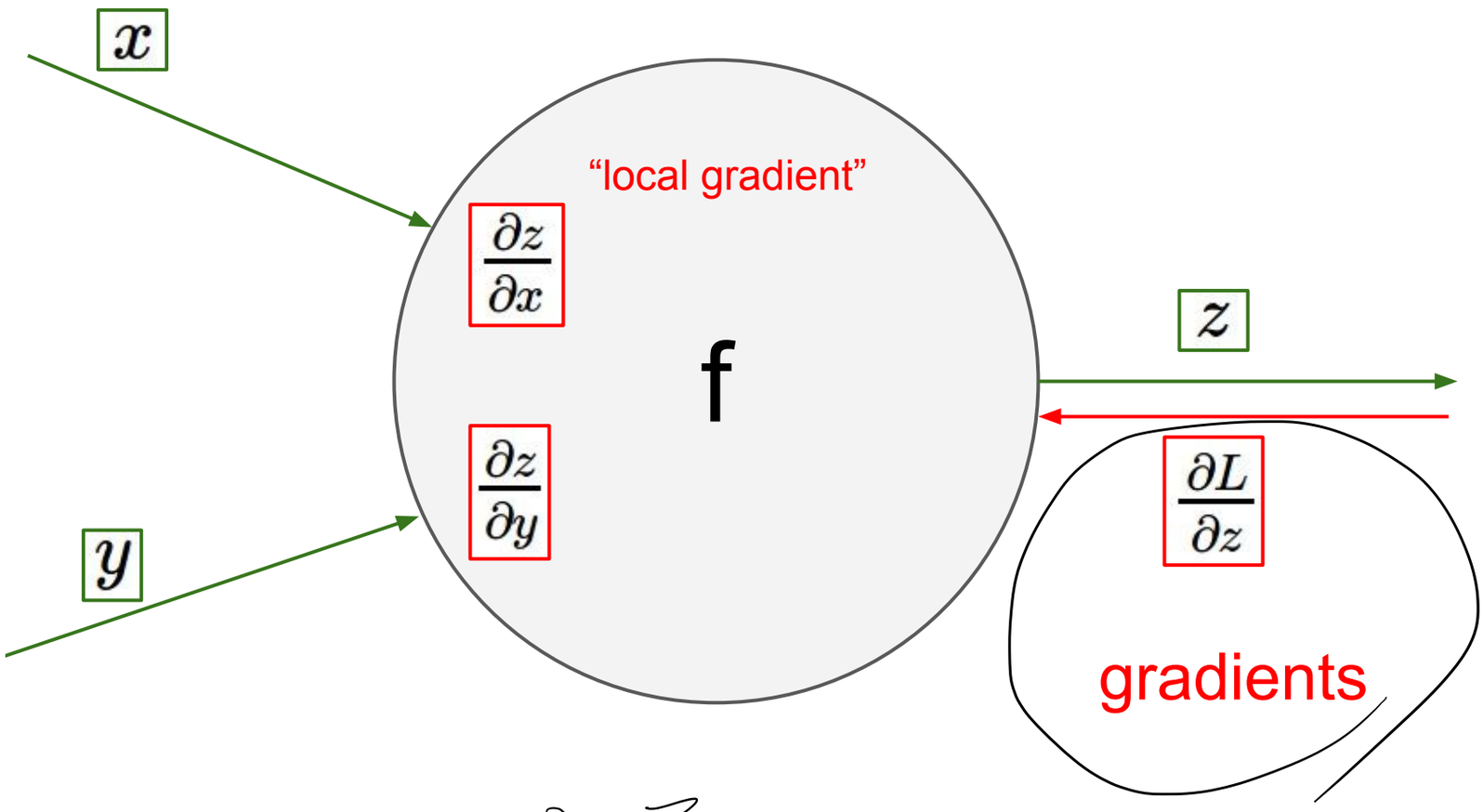
$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial z} \times \frac{\partial z}{\partial w_i}$$



$$f = \sigma(w^T \cdot x + b)$$

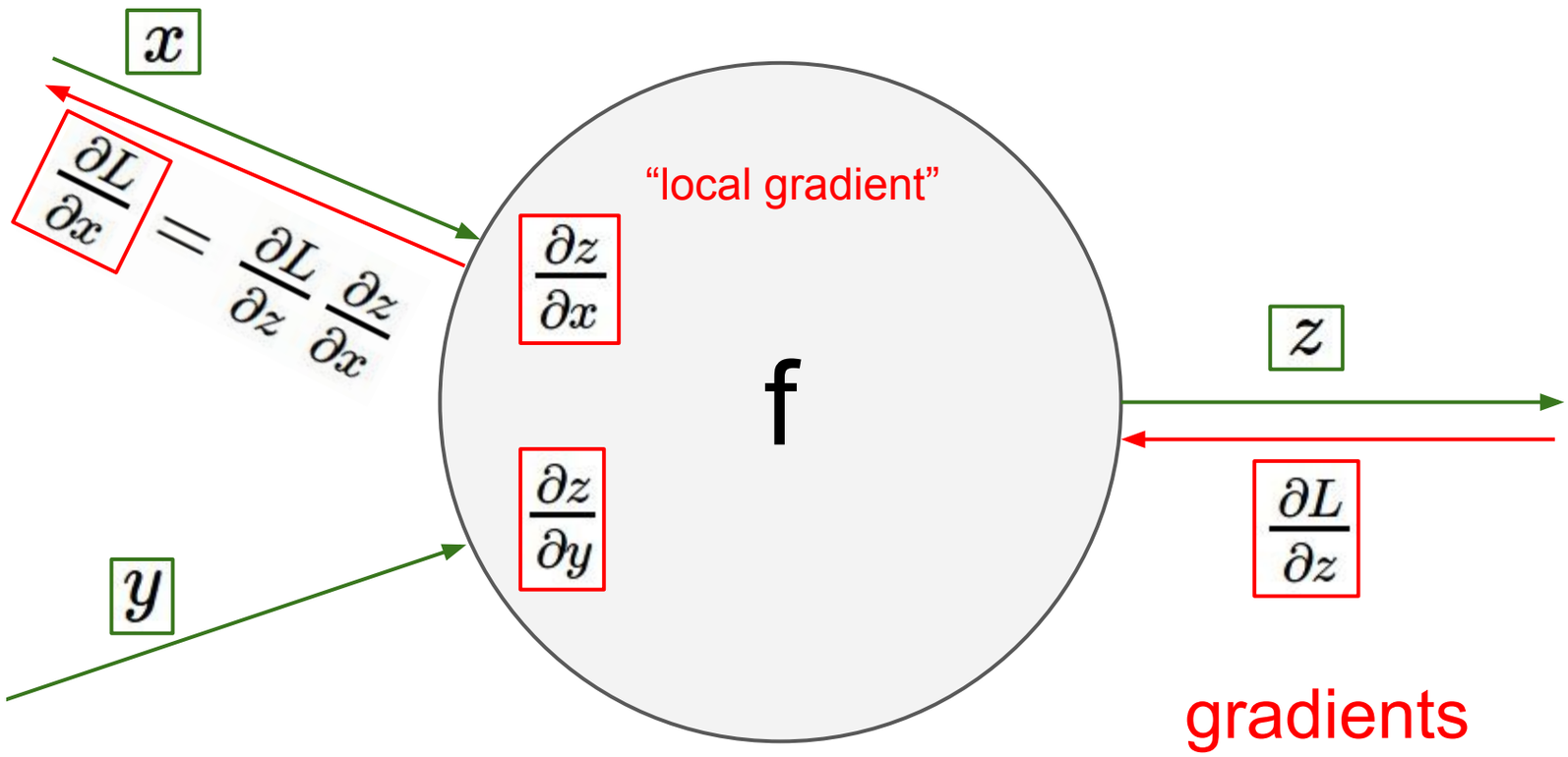
$$\frac{\partial f}{\partial x_1} = z, \quad \frac{\partial f}{\partial x_2} = z, \quad \frac{\partial f}{\partial w_i} = z$$

Backpropagation

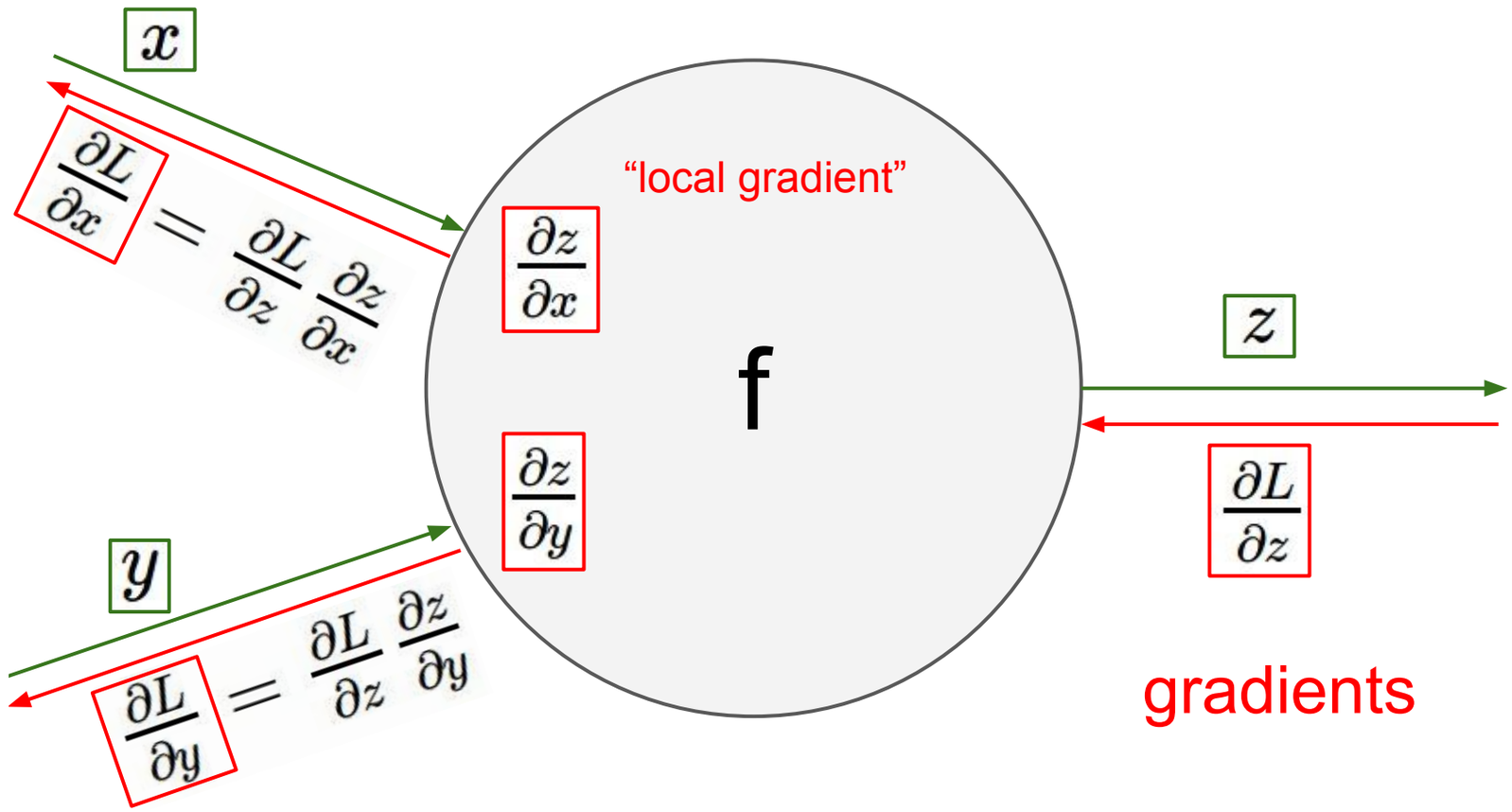


$$\frac{\partial L}{\partial z} \times \frac{\partial z}{\partial w_i}$$

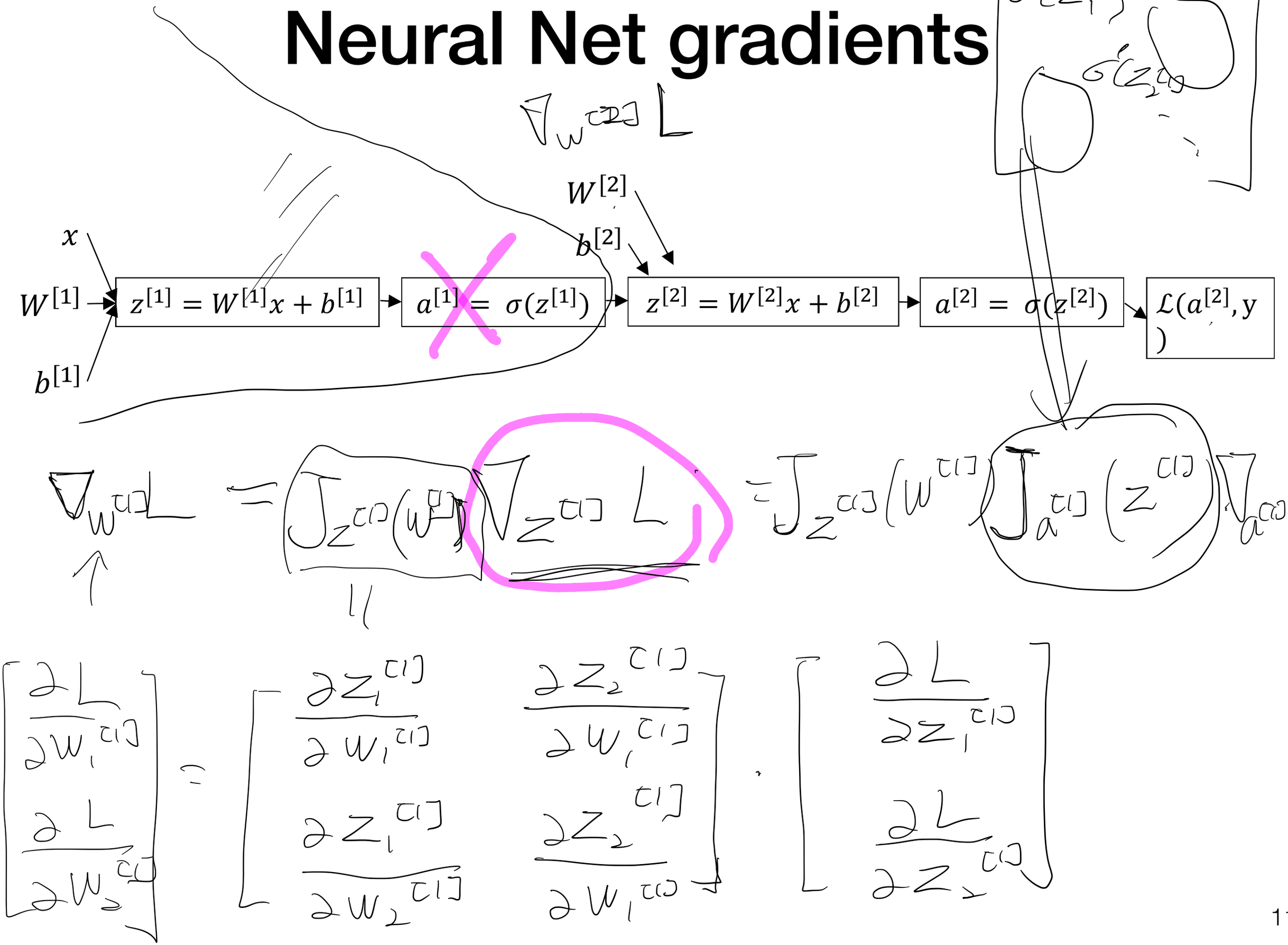
Backpropagation



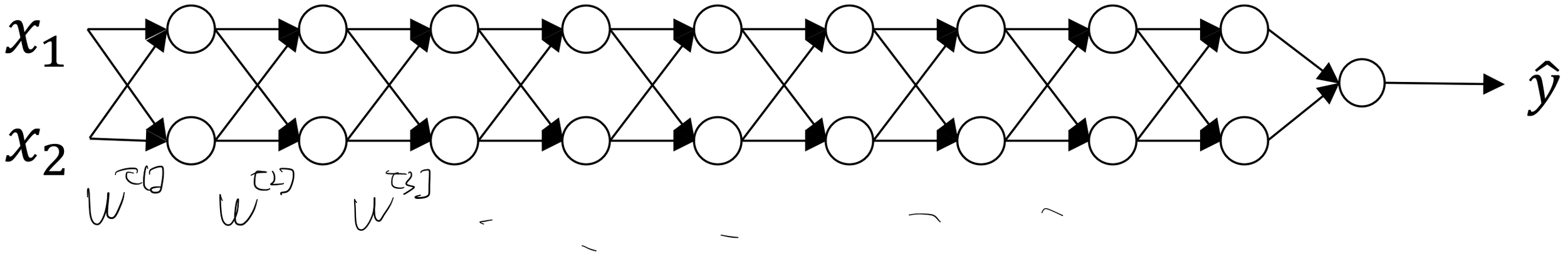
Backpropagation



Neural Net gradients



Vanishing/exploding gradients



$\nabla_{w^{[1]}} L$

$A = X_1 X_2 X_3 X_4 \cdots X_{10}$

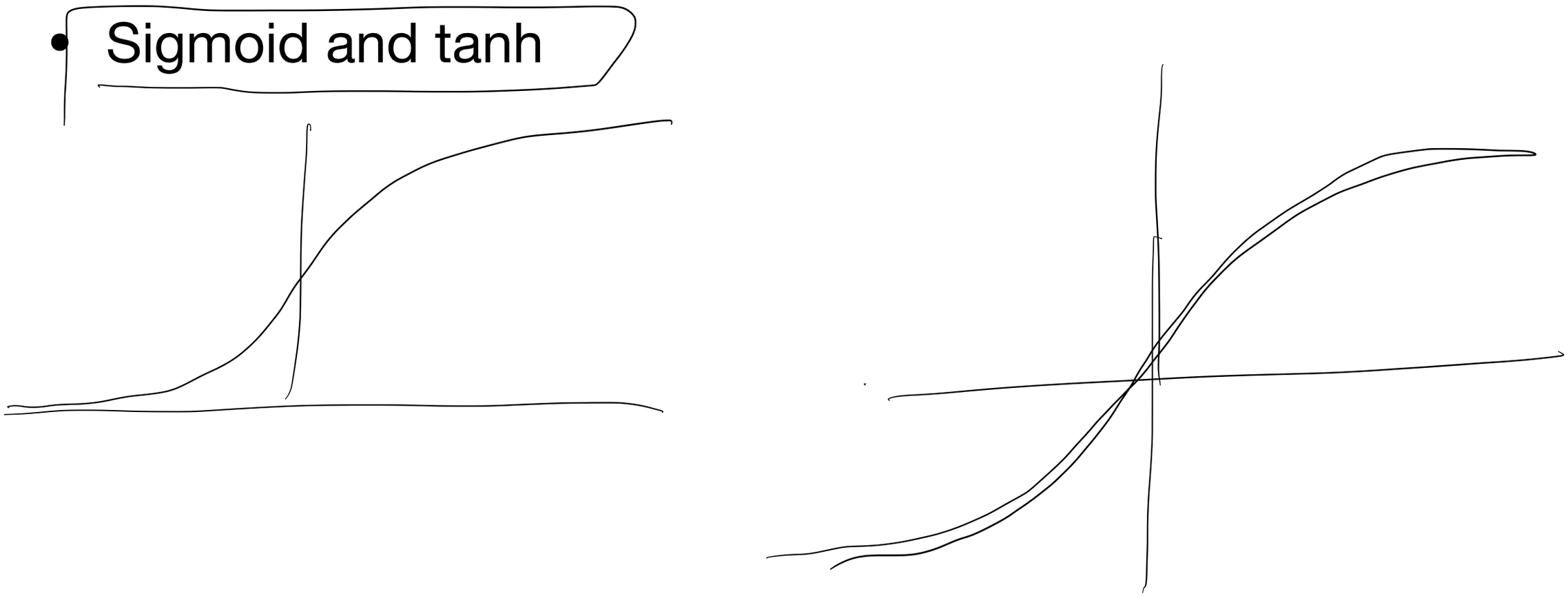
$L^1 \quad L^1 \quad\quad L^1$

$\|A\|_2 \leq \|X_1\|_2 \|X_2\|_2 \cdots \|X_{10}\|_2$

$> 1 \quad > 1 \quad\quad > 1$

Derivatives of activation function

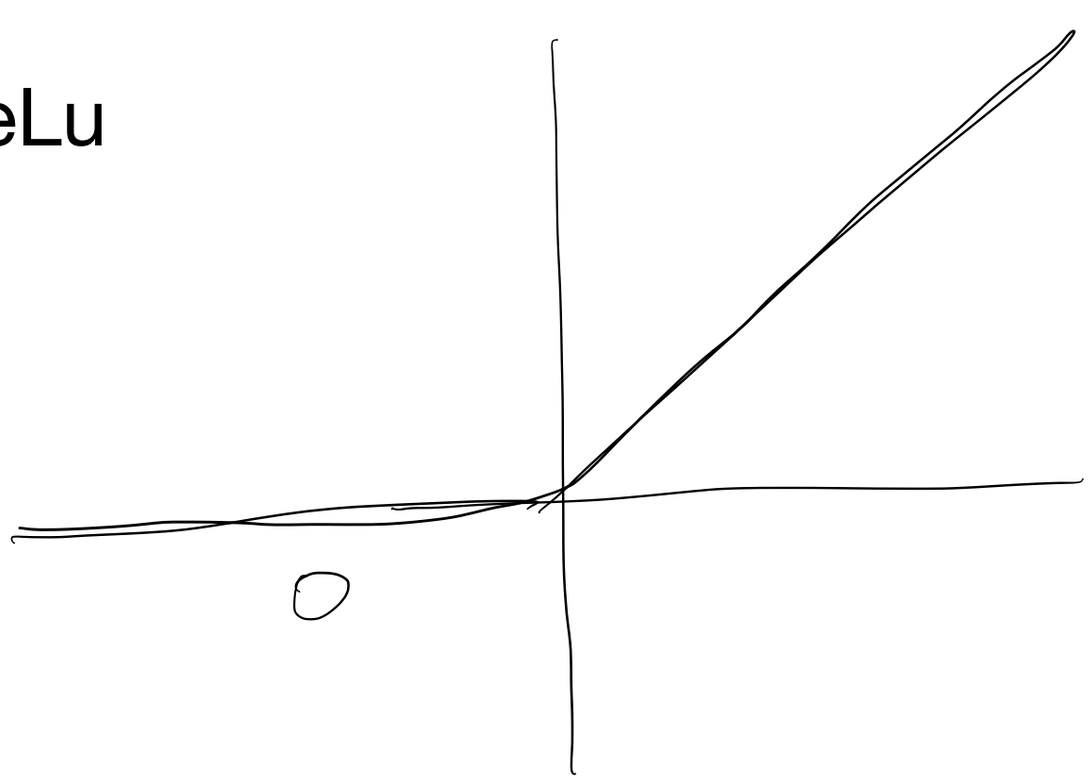
- Sigmoid and tanh



$$J_{a^{[l]}}(z^{[l]}) = \begin{bmatrix} g'(z_1^{[l]}) & & 0 \\ & \ddots & \\ 0 & & g'(z_n^{[l]}) \end{bmatrix}$$

Derivatives of activation function

- ReLu



$g'(z) = 0$ or 1

$\| \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \|_2 = 1$