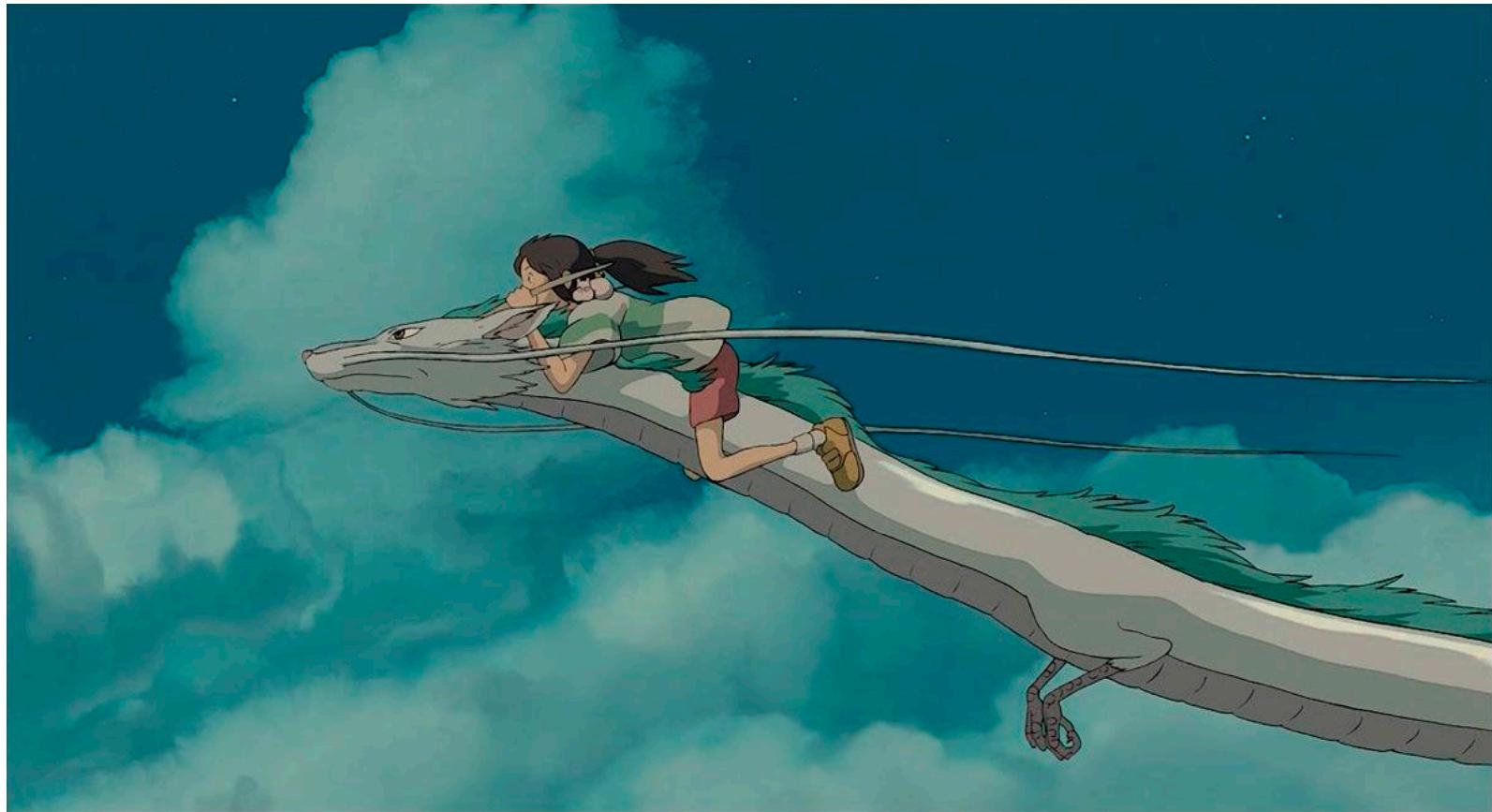
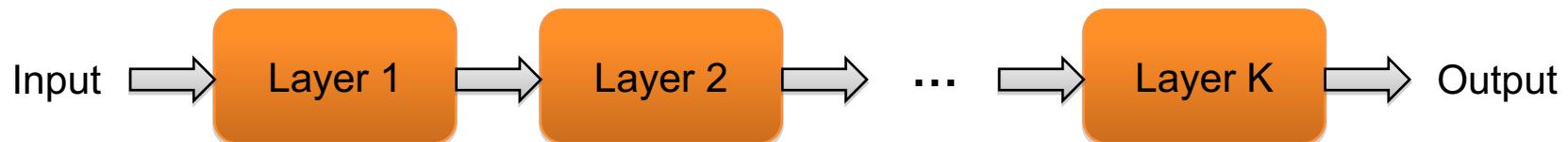


How to train a multi-layer network?

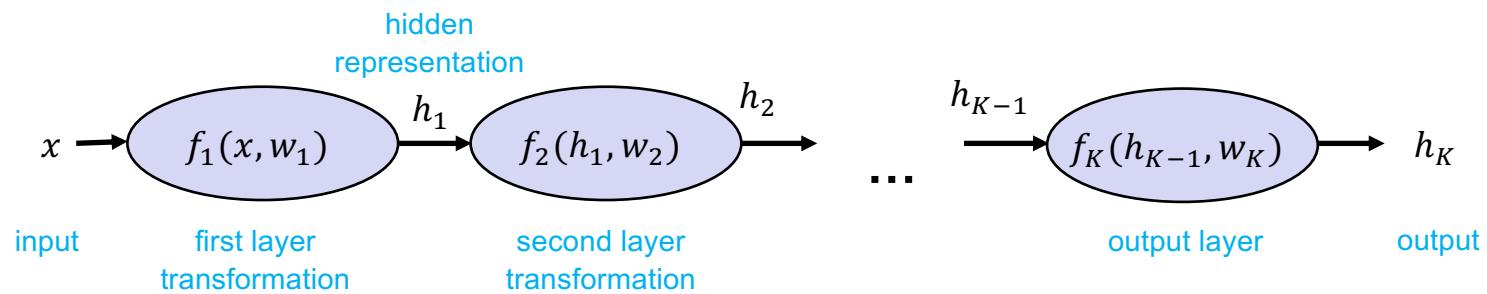


Recall: Multi-layer neural networks

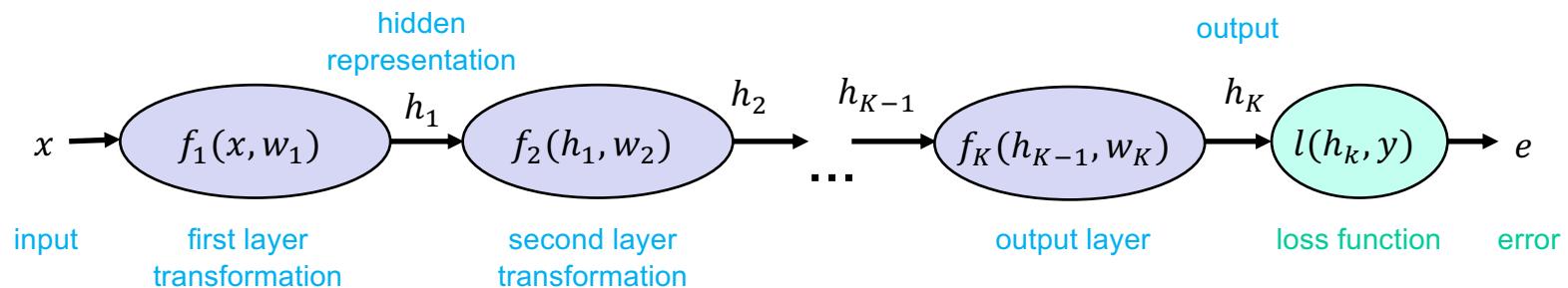
- The function computed by the network is a composition of the functions computed by individual layers (e.g., linear layers and nonlinearities):



- More precisely:



Training a multi-layer network

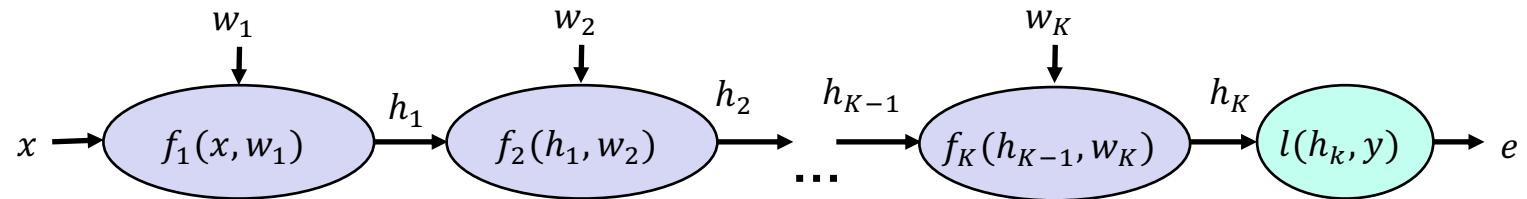


- What is the SGD update for the parameters w_k of the k th layer?

$$w_k \leftarrow w_k - \eta \frac{\partial e}{\partial w_k}$$

- To train the network, we need to find the gradient of the error w.r.t. the parameters of each layer, $\frac{\partial e}{\partial w_k}$

Computation graph

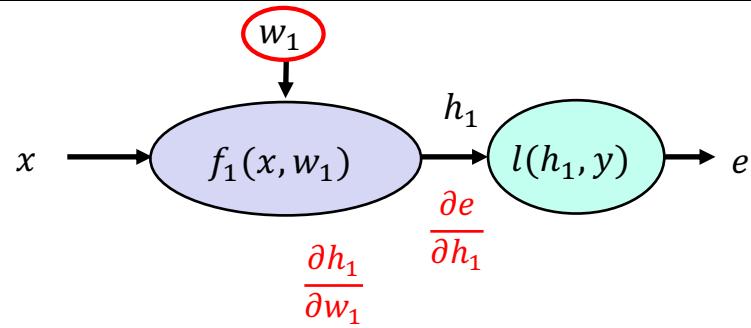


Chain rule

Let's start with $k = 1$

$$e = l(f_1(x, w_1), y)$$

$$\frac{\partial}{\partial w_1} l(f_1(x, w_1), y) =$$



Example: $e = (y - w_1^T x)^2$

$$h_1 = f_1(x, w_1) = w_1^T x$$

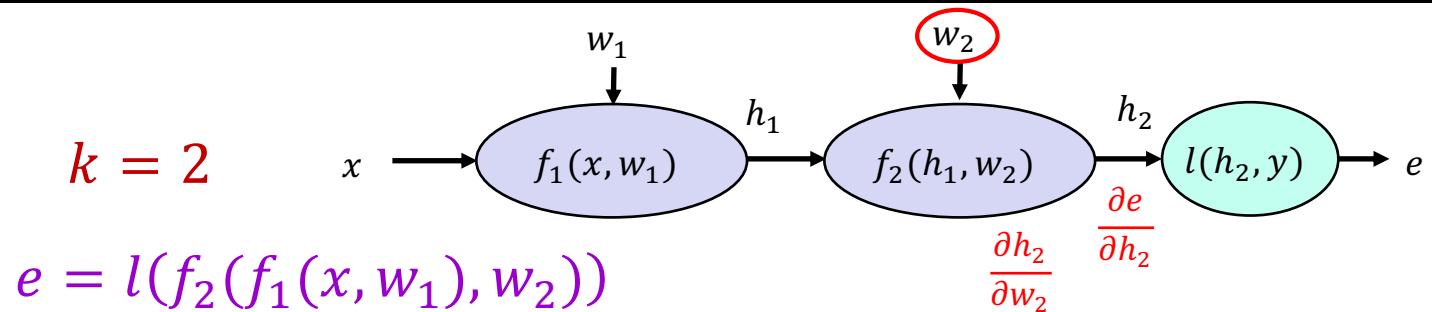
$$\frac{\partial h_1}{\partial w_1} =$$

$$e = l(h_1, y) = (y - h_1)^2$$

$$\frac{\partial e}{\partial h_1} =$$

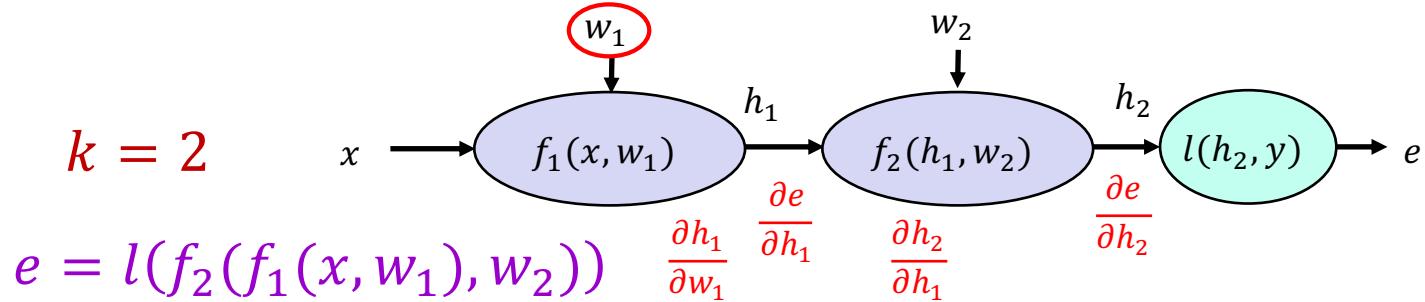
$$\frac{\partial e}{\partial w_1} = \frac{\partial e}{\partial h_1} \frac{\partial h_1}{\partial w_1}$$

Chain rule



$$\frac{\partial e}{\partial w_2} =$$

Chain rule



$$\frac{\partial e}{\partial w_2} = \frac{\partial e}{\partial h_2} \frac{\partial h_2}{\partial w_2}$$



Example: $e = -\log(\sigma(w_1^T x))$ (assume $y = 1$)

$$h_1 = f_1(x, w_1) = w_1^T x$$

$$\frac{\partial h_1}{\partial w_1} =$$

$$h_2 = f_2(h_1) = \sigma(h_1)$$

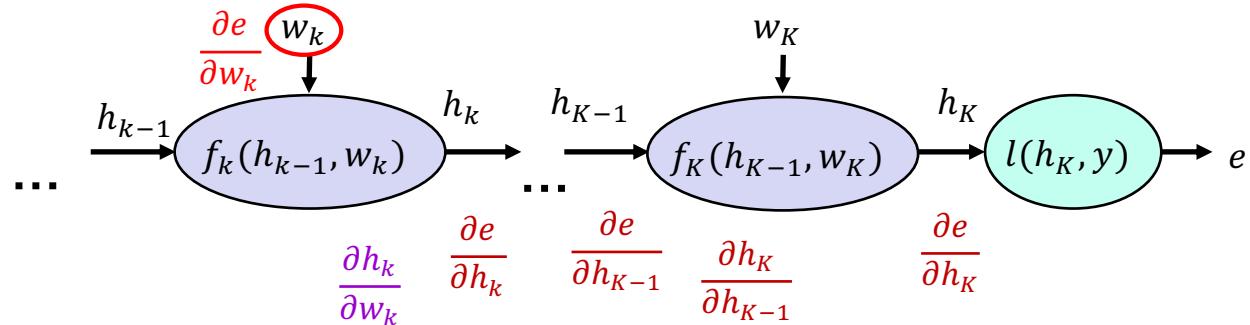
$$\frac{\partial h_2}{\partial h_1} =$$

$$e = l(h_2, 1) = -\log(h_2)$$

$$\frac{\partial e}{\partial h_2} =$$

$$\frac{\partial e}{\partial w_1} = \frac{\partial e}{\partial h_2} \frac{\partial h_2}{\partial h_1} \frac{\partial h_1}{\partial w_1} =$$

Chain rule



General case:

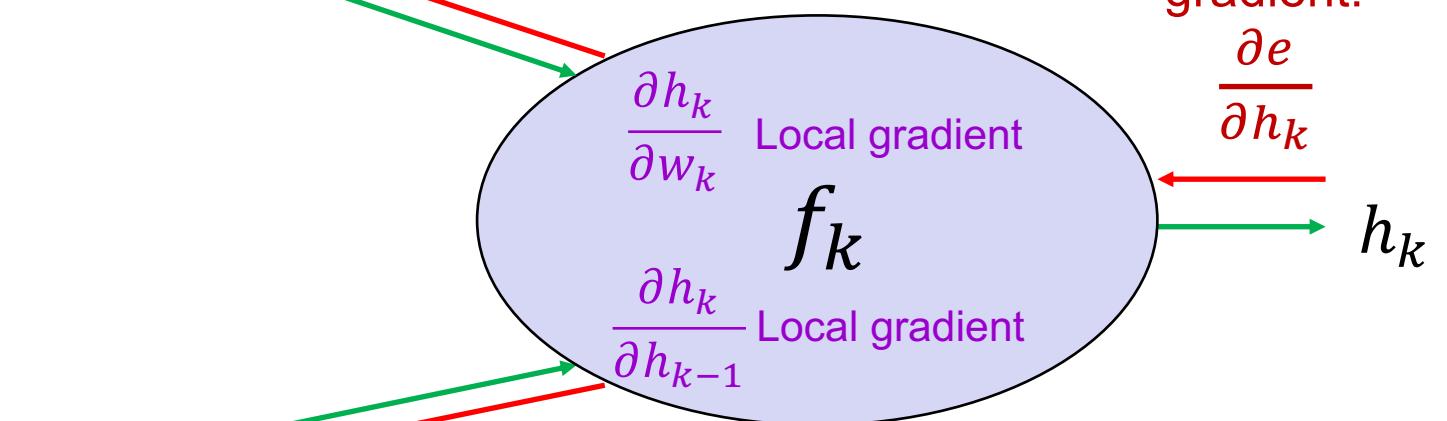
$$\frac{\partial e}{\partial w_k} = \boxed{\frac{\partial e}{\partial h_K} \frac{\partial h_K}{\partial h_{K-1}} \dots \frac{\partial h_{k+1}}{\partial h_k}} \frac{\partial h_k}{\partial w_k}$$

Upstream gradient Local
 $\frac{\partial e}{\partial h_k}$ gradient

Backpropagation summary

Parameter update:

$$\frac{\partial e}{\partial w_k} = \frac{\partial e}{\partial h_k} \frac{\partial h_k}{\partial w_k}$$



Upstream gradient:

$$\frac{\partial e}{\partial h_k}$$

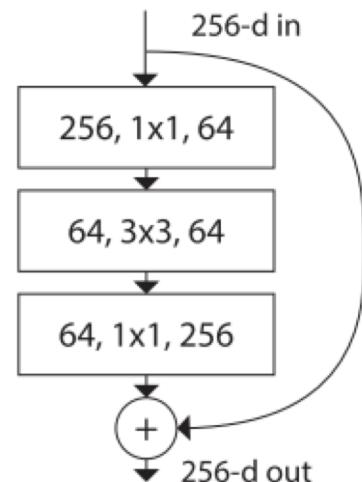
→ Forward pass
← Backward pass

Upstream gradient:

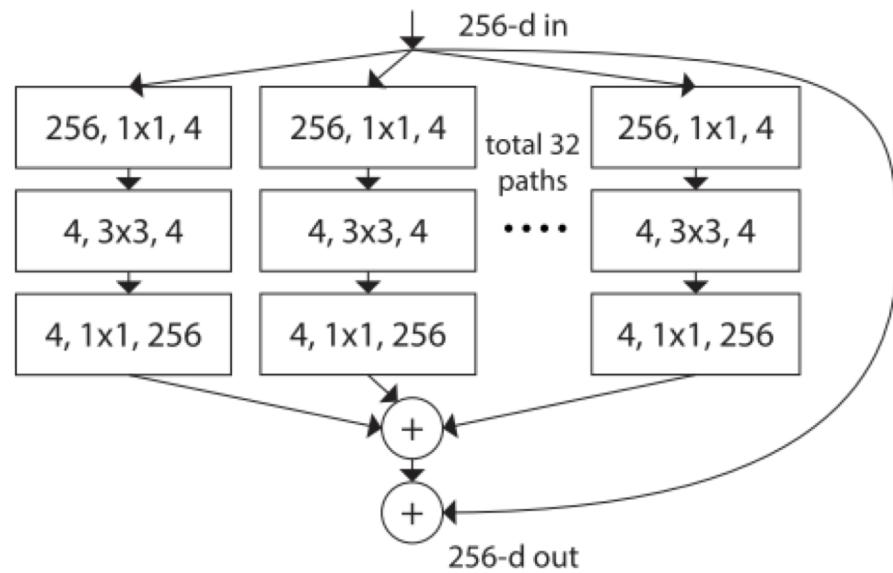
$$\frac{\partial e}{\partial h_{k-1}} = \frac{\partial e}{\partial h_k} \frac{\partial h_k}{\partial h_{k-1}}$$

What about more general computation graphs?

ResNet

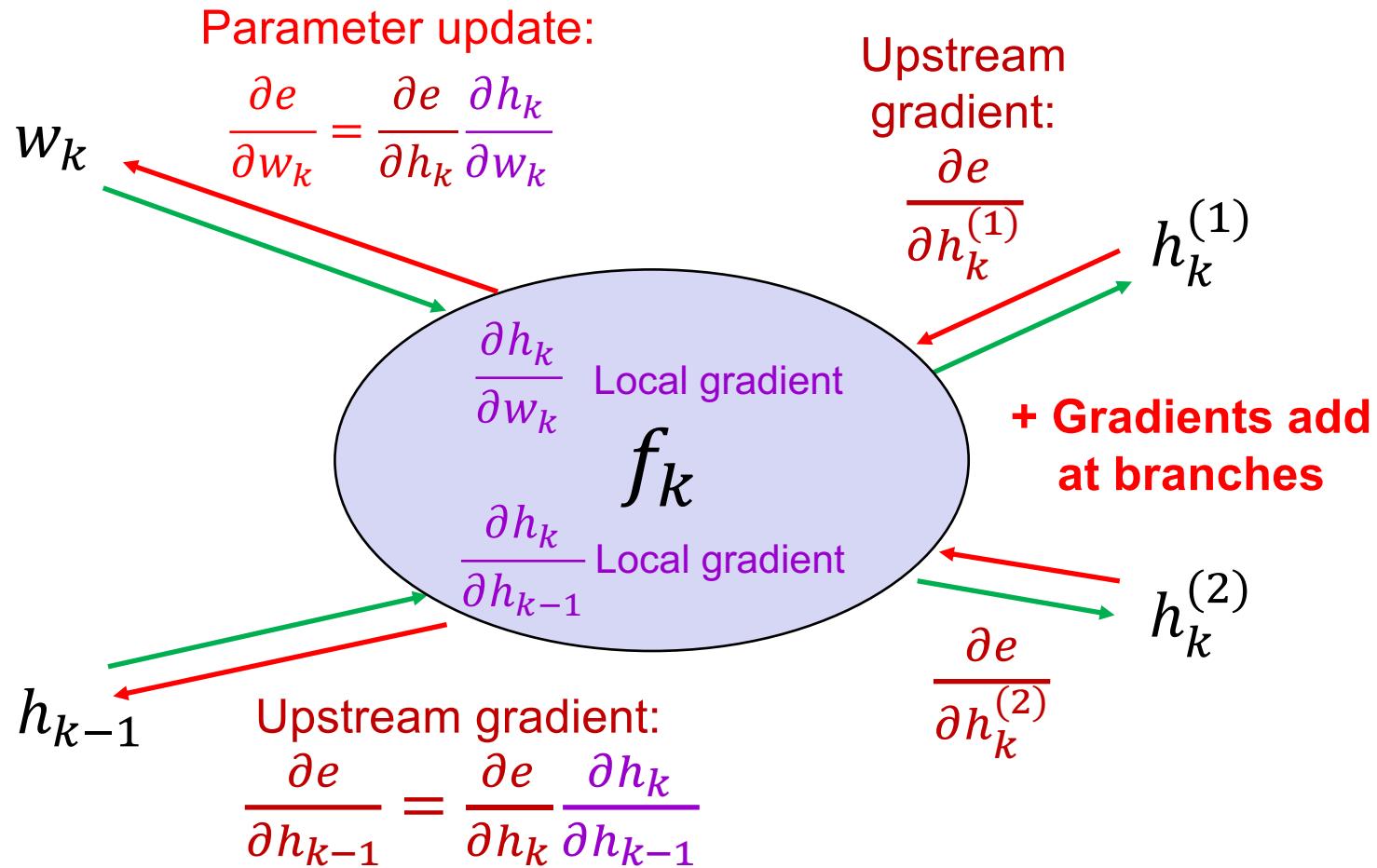


ResNeXt



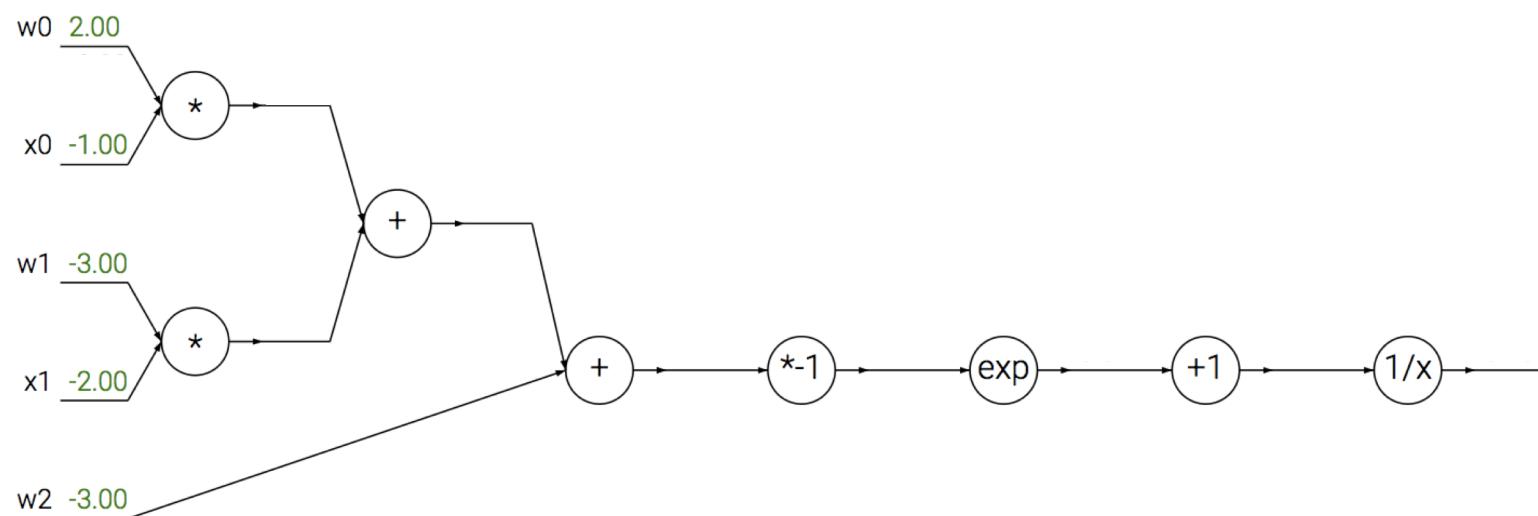
[Figure source](#)

What about more general computation graphs?



A detailed example

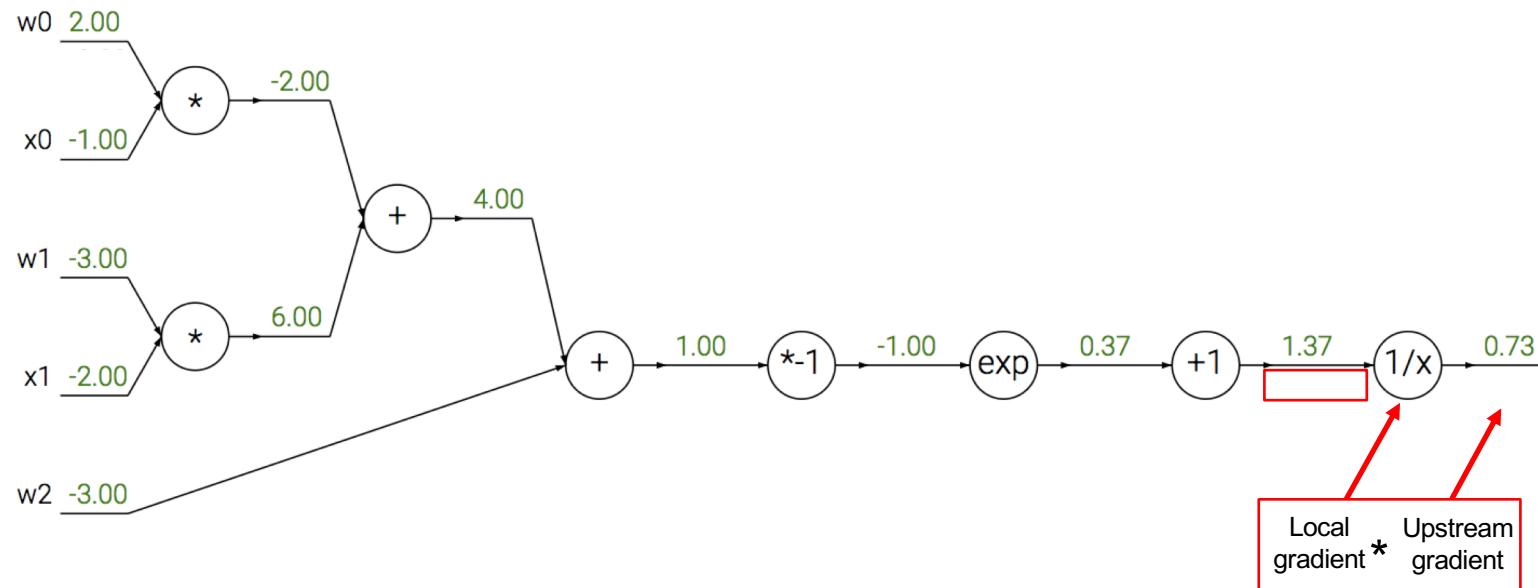
$$f(x, w) = \frac{1}{1 + \exp[-(w_0 x_0 + w_1 x_1 + w_2)]}$$



Source: [Stanford 231n](#)

A detailed example

$$f(x, w) = \frac{1}{1 + \exp[-(w_0 x_0 + w_1 x_1 + w_2)]}$$



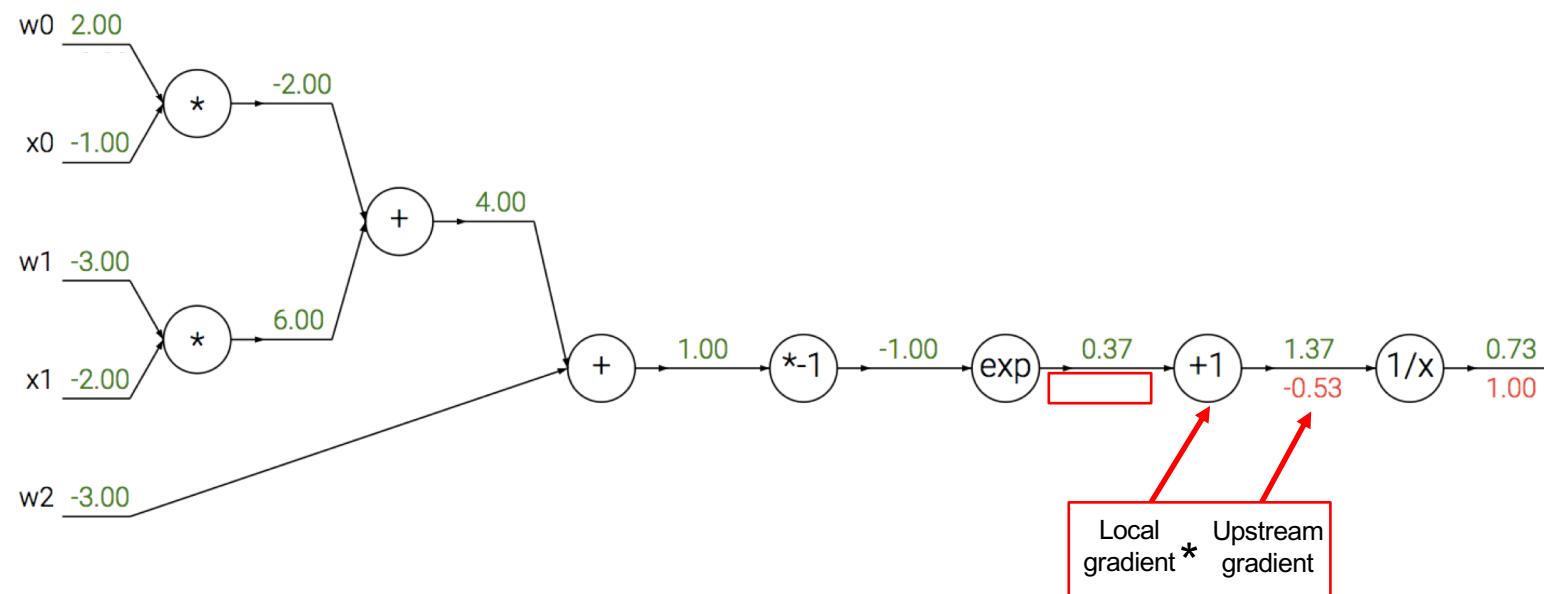
$$(1/x)' = -1/x^2$$

$$-\frac{1}{1.37^2} * 1 = -0.53$$

Source: [Stanford 231n](#)

A detailed example

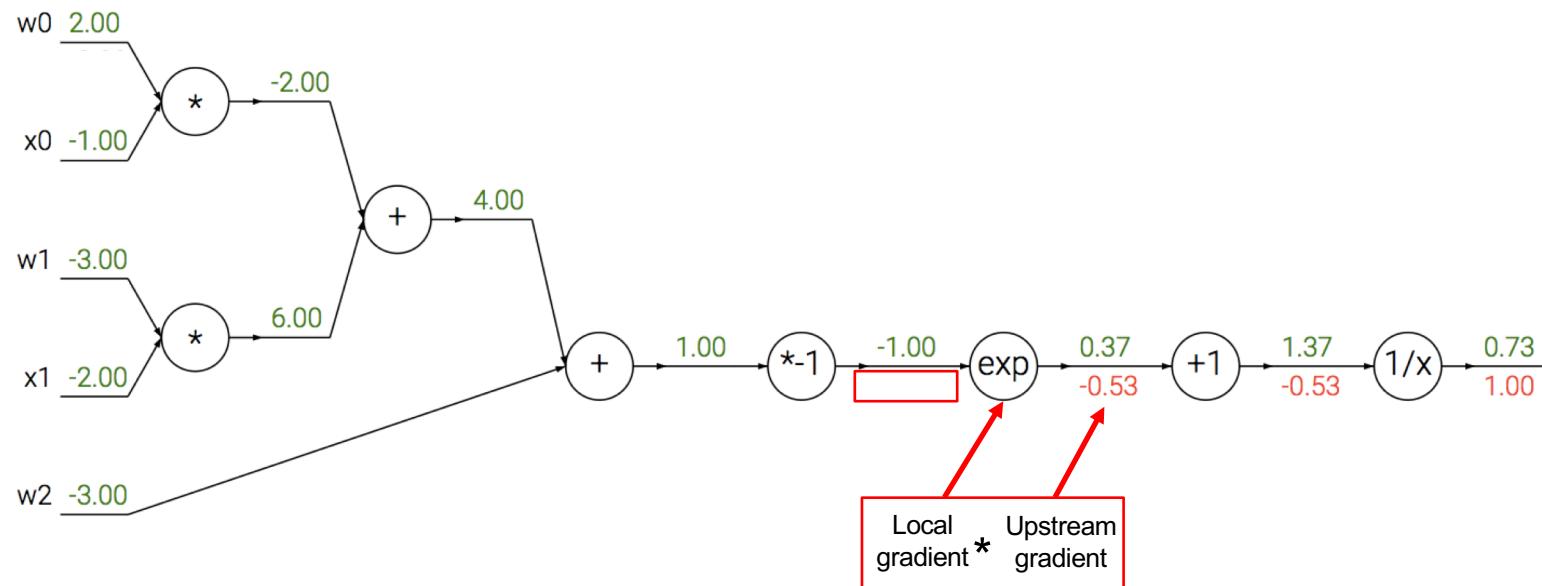
$$f(x, w) = \frac{1}{1 + \exp[-(w_0 x_0 + w_1 x_1 + w_2)]}$$



Source: [Stanford 231n](#)

A detailed example

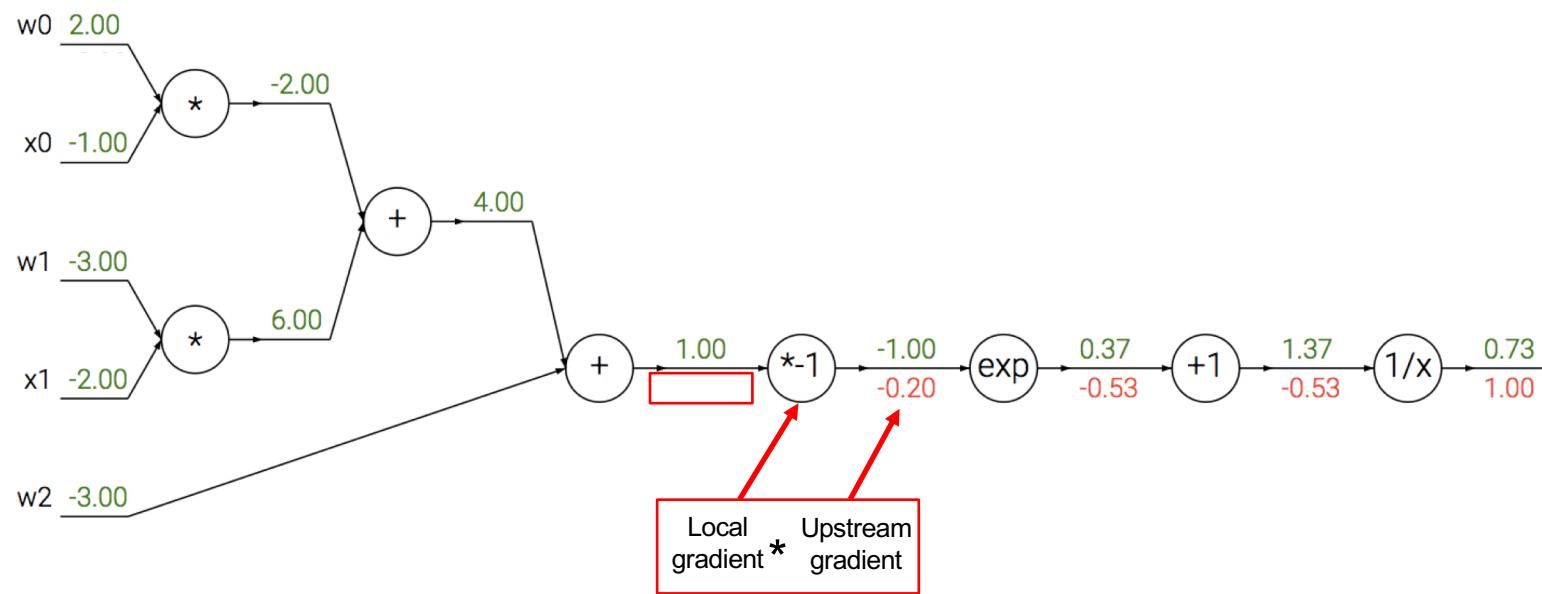
$$f(x, w) = \frac{1}{1 + \exp[-(w_0 x_0 + w_1 x_1 + w_2)]}$$



Source: [Stanford 231n](#)

A detailed example

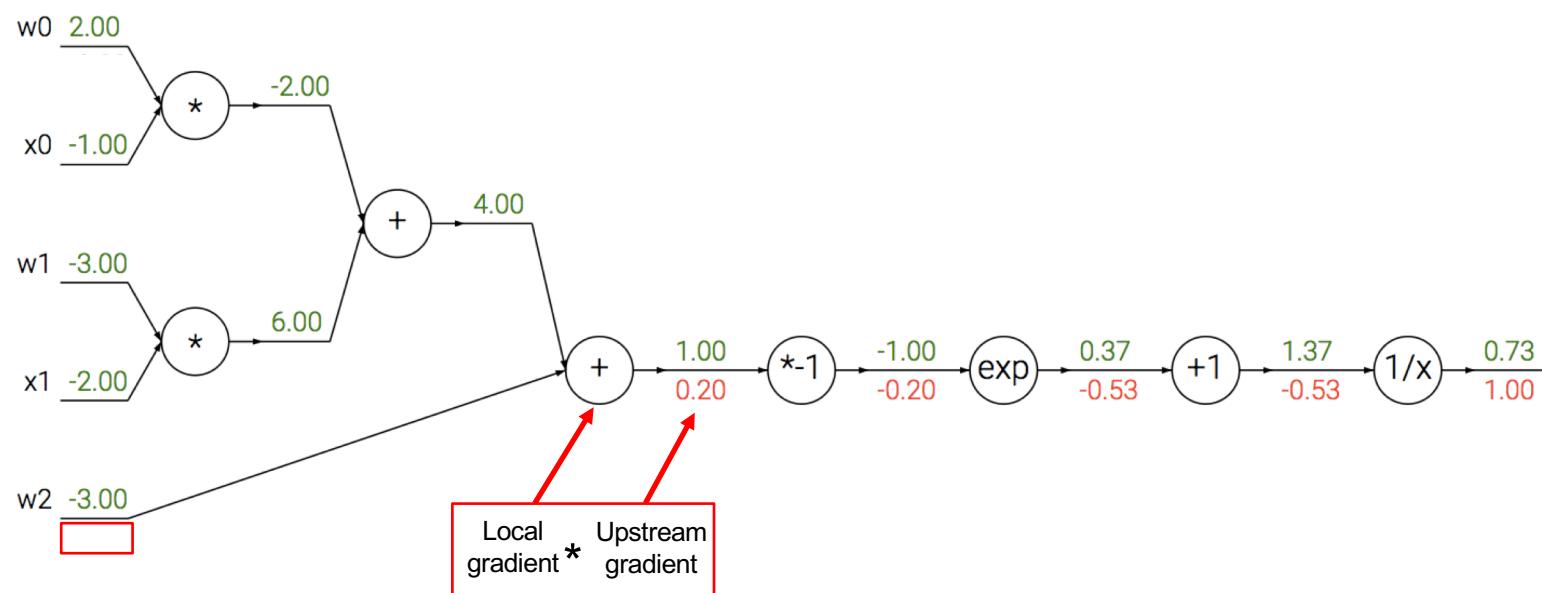
$$f(x, w) = \frac{1}{1 + \exp[-(w_0 x_0 + w_1 x_1 + w_2)]}$$



Source: [Stanford 231n](#)

A detailed example

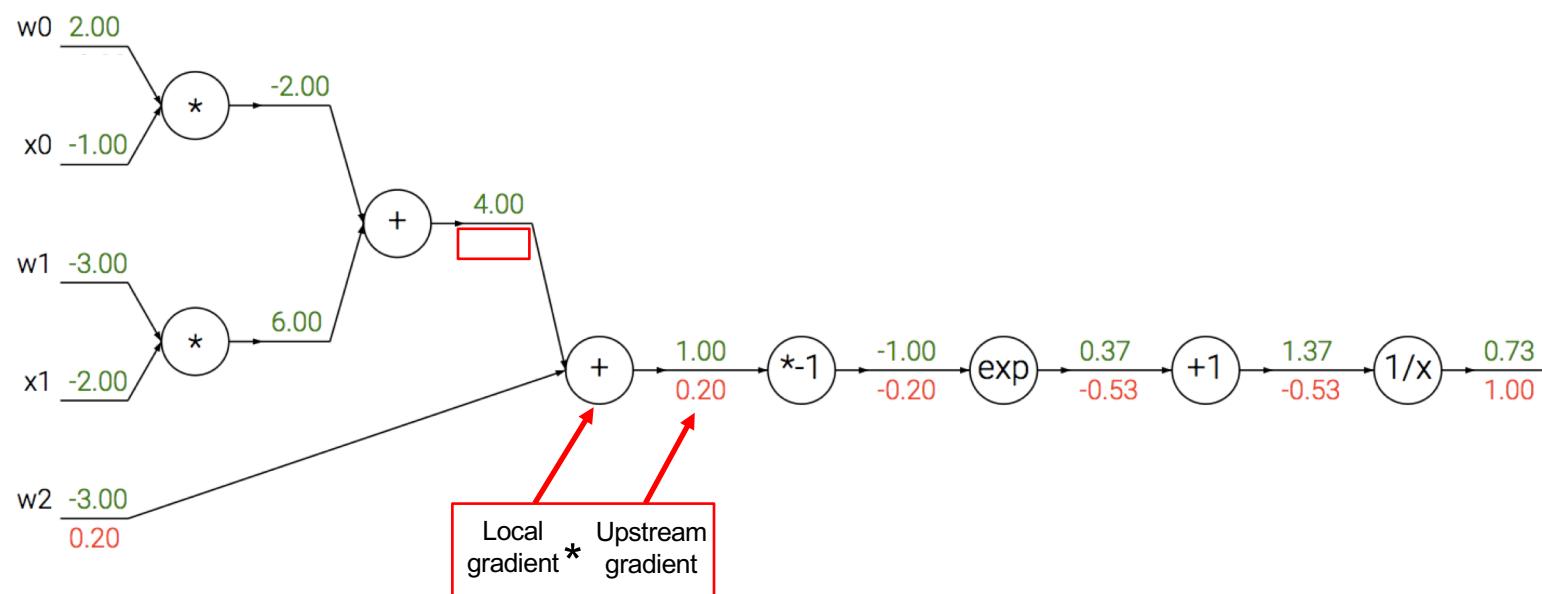
$$f(x, w) = \frac{1}{1 + \exp[-(w_0 x_0 + w_1 x_1 + w_2)]}$$



Source: [Stanford 231n](#)

A detailed example

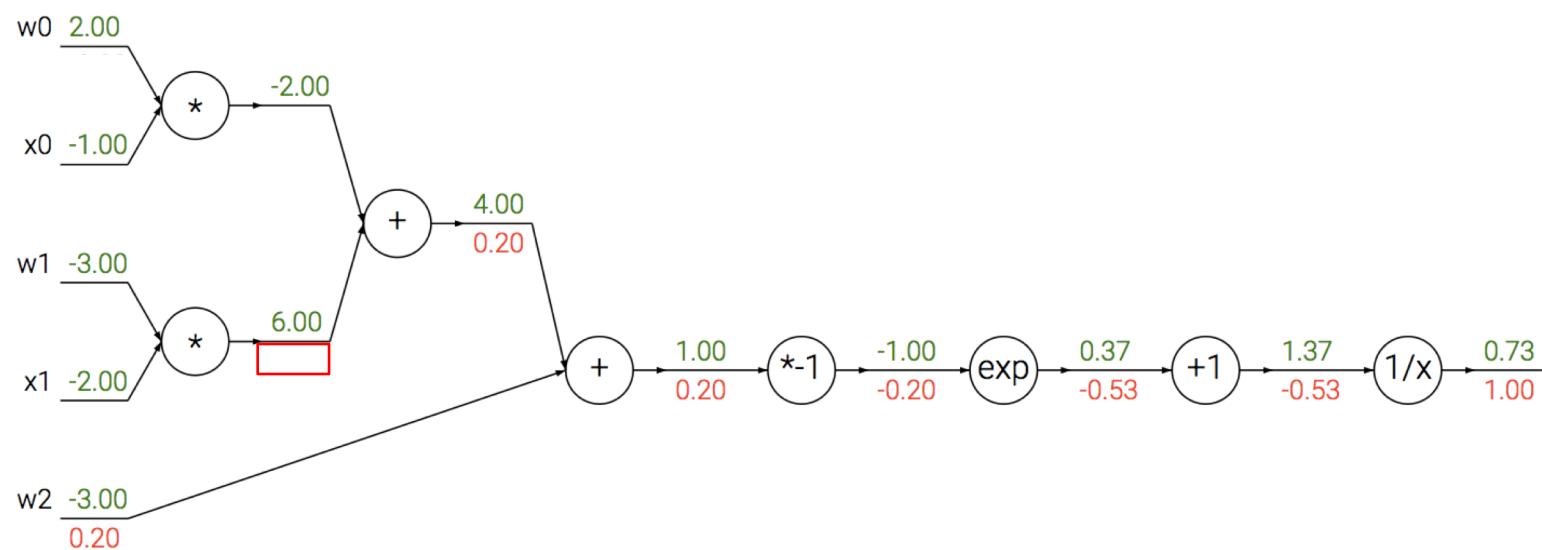
$$f(x, w) = \frac{1}{1 + \exp[-(w_0 x_0 + w_1 x_1 + w_2)]}$$



Source: [Stanford 231n](#)

A detailed example

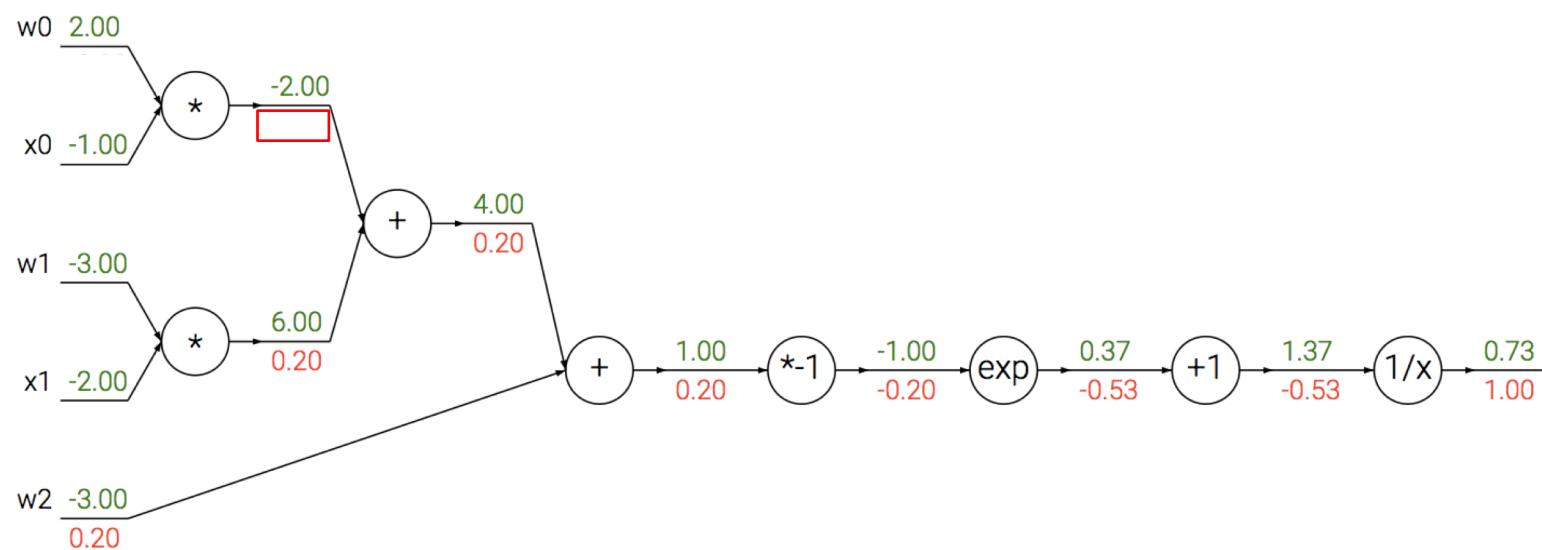
$$f(x, w) = \frac{1}{1 + \exp[-(w_0 x_0 + w_1 x_1 + w_2)]}$$



Source: [Stanford 231n](#)

A detailed example

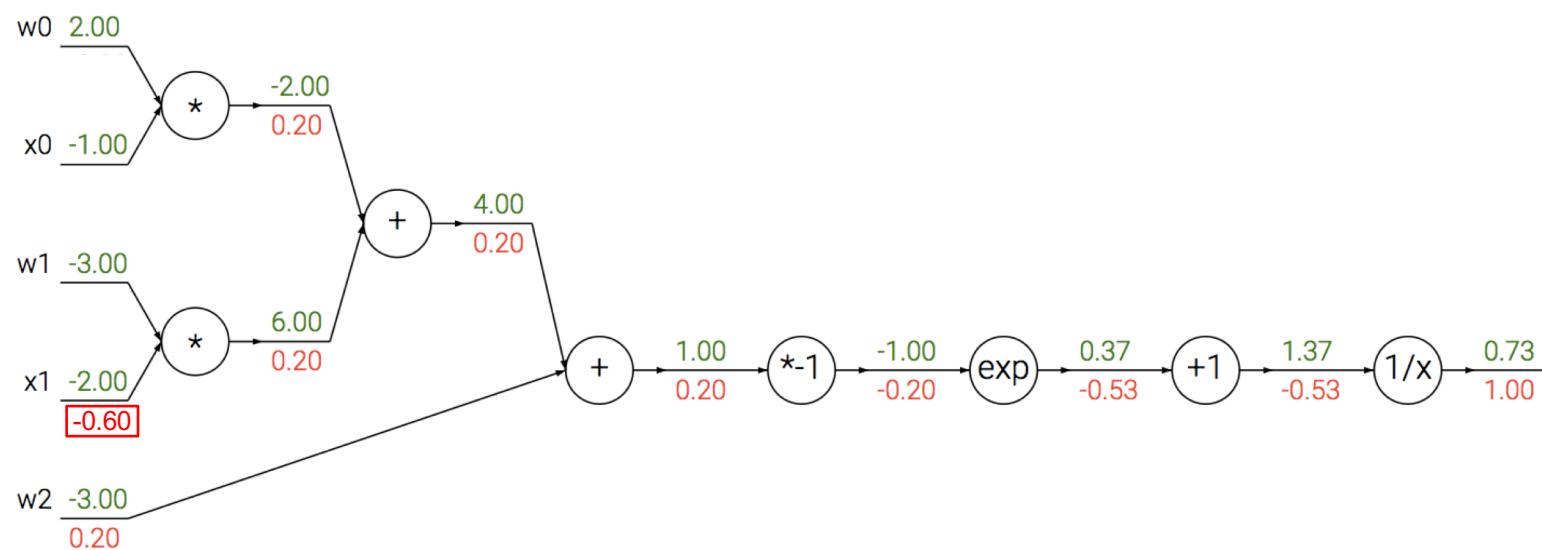
$$f(x, w) = \frac{1}{1 + \exp[-(w_0x_0 + w_1x_1 + w_2)]}$$



Source: [Stanford 231n](#)

A detailed example

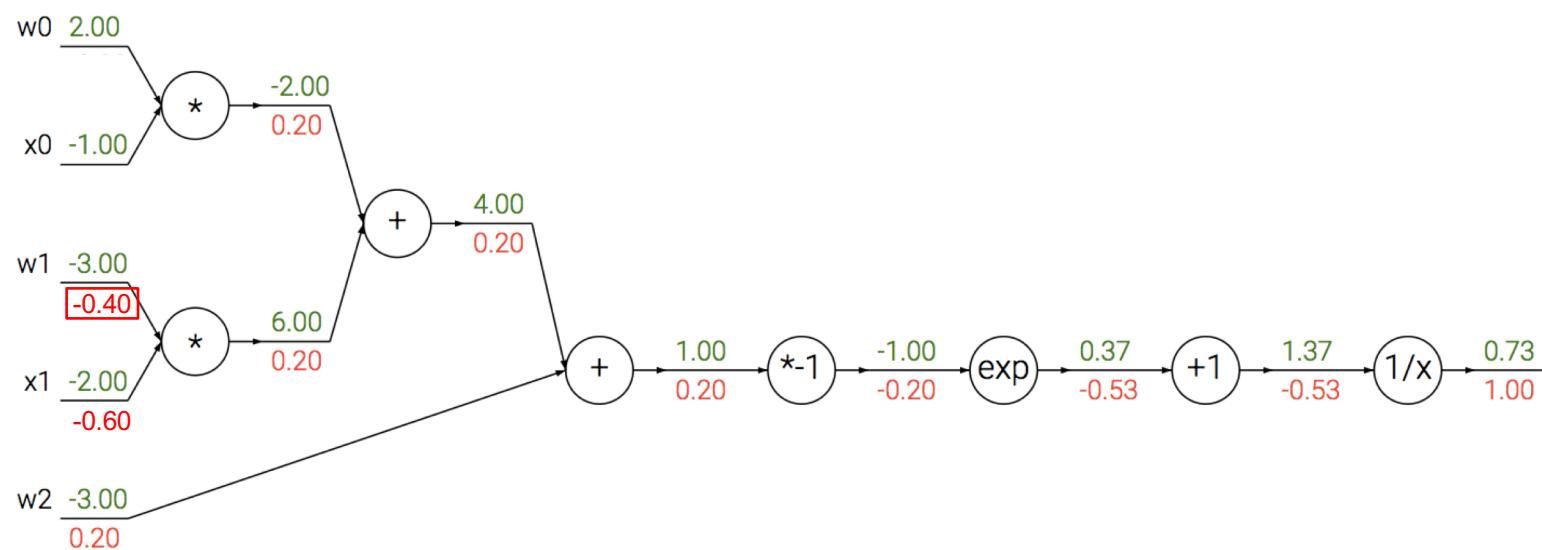
$$f(x, w) = \frac{1}{1 + \exp[-(w_0x_0 + w_1x_1 + w_2)]}$$



Source: [Stanford 231n](#)

A detailed example

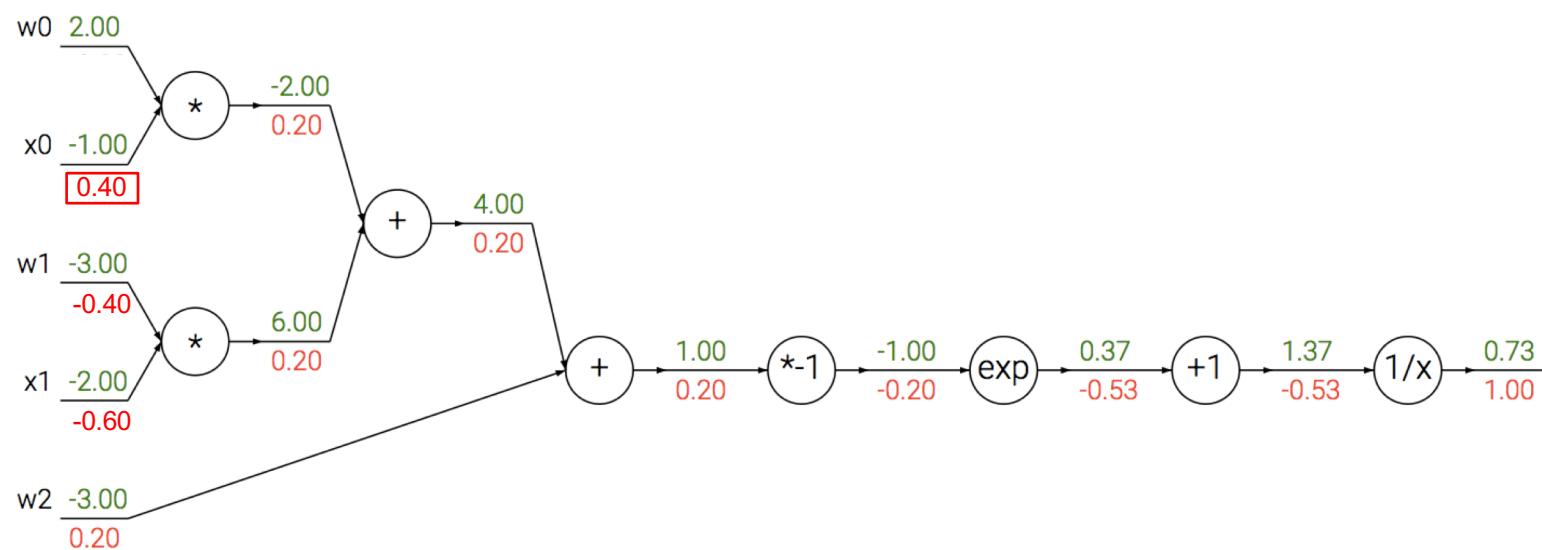
$$f(x, w) = \frac{1}{1 + \exp[-(w_0x_0 + w_1x_1 + w_2)]}$$



Source: [Stanford 231n](#)

A detailed example

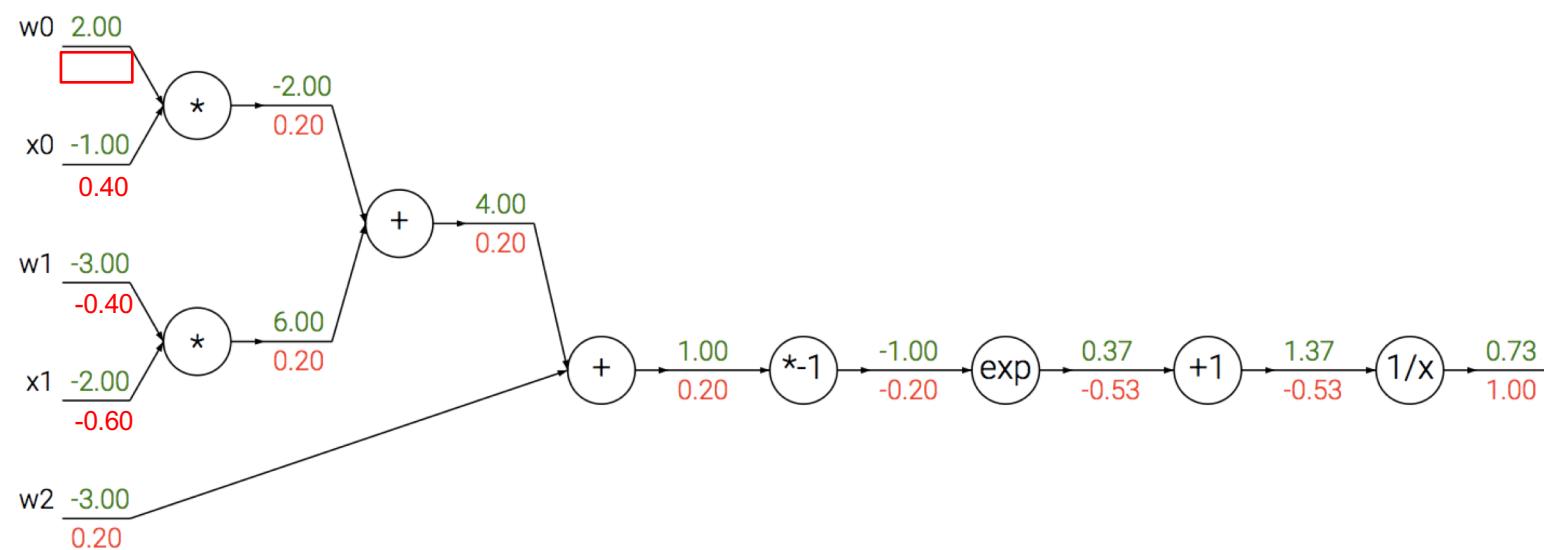
$$f(x, w) = \frac{1}{1 + \exp[-(w_0x_0 + w_1x_1 + w_2)]}$$



Source: [Stanford 231n](#)

A detailed example

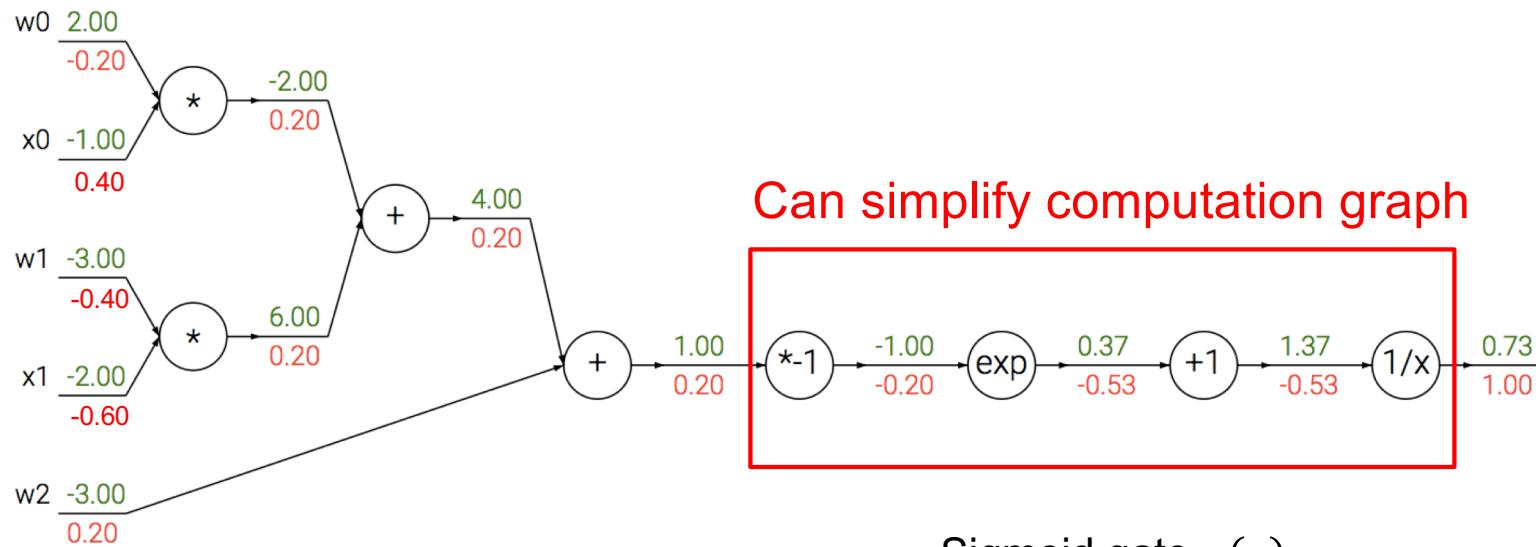
$$f(x, w) = \frac{1}{1 + \exp[-(w_0x_0 + w_1x_1 + w_2)]}$$



Source: [Stanford 231n](#)

A detailed example

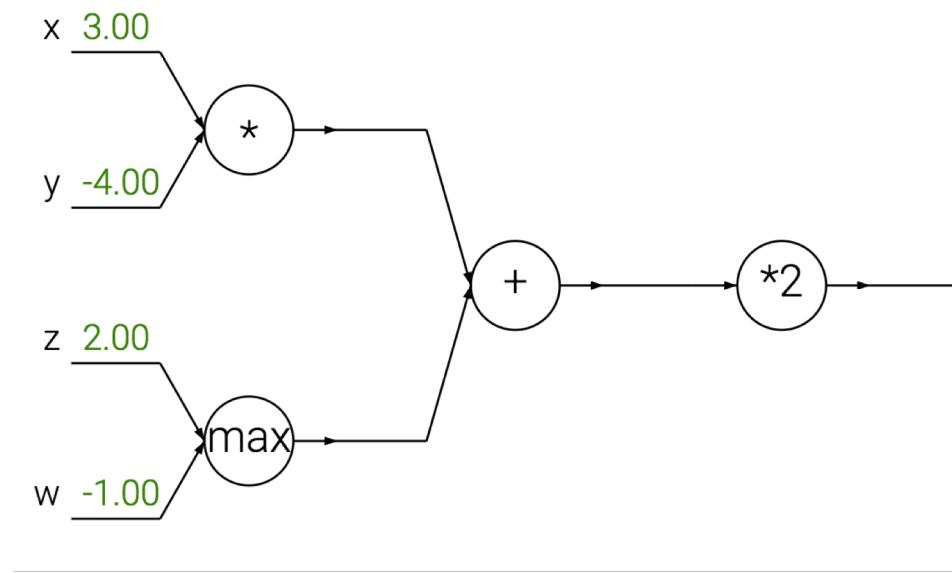
$$f(x, w) = \frac{1}{1 + \exp[-(w_0x_0 + w_1x_1 + w_2)]}$$



Sigmoid gate $\sigma(x)$
 $\sigma'(x) = \sigma(x)(1 - \sigma(x))$
 $\sigma(1)(1 - \sigma(1)) = 0.73 * (1 - 0.73) = 0.20$

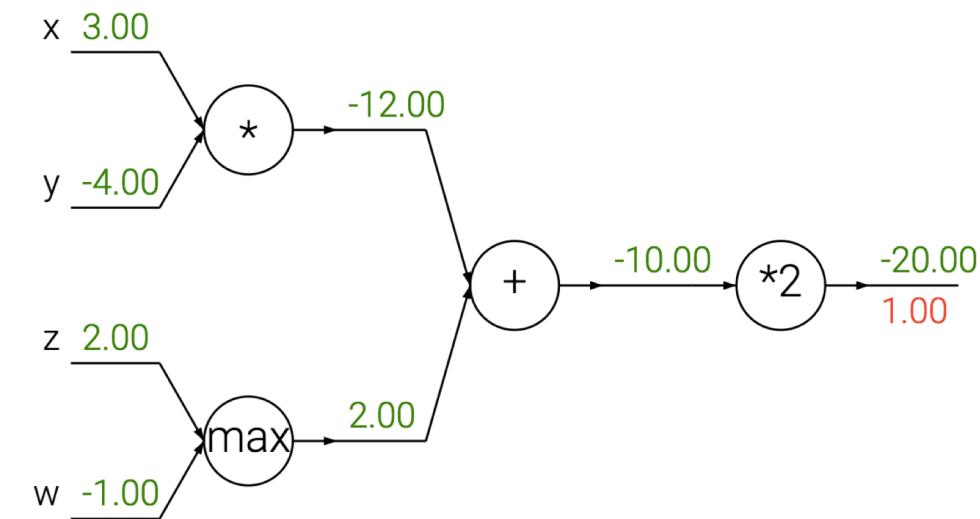
Source: [Stanford 231n](#)

Patterns in gradient flow



Source: [Stanford 231n](#)

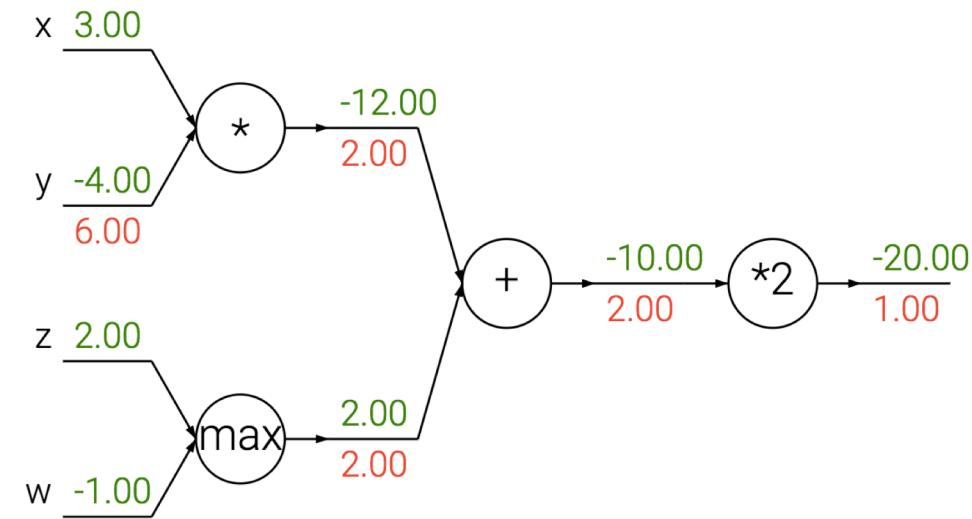
Patterns in gradient flow



Add gate: “gradient distributor”

Source: [Stanford 231n](#)

Patterns in gradient flow

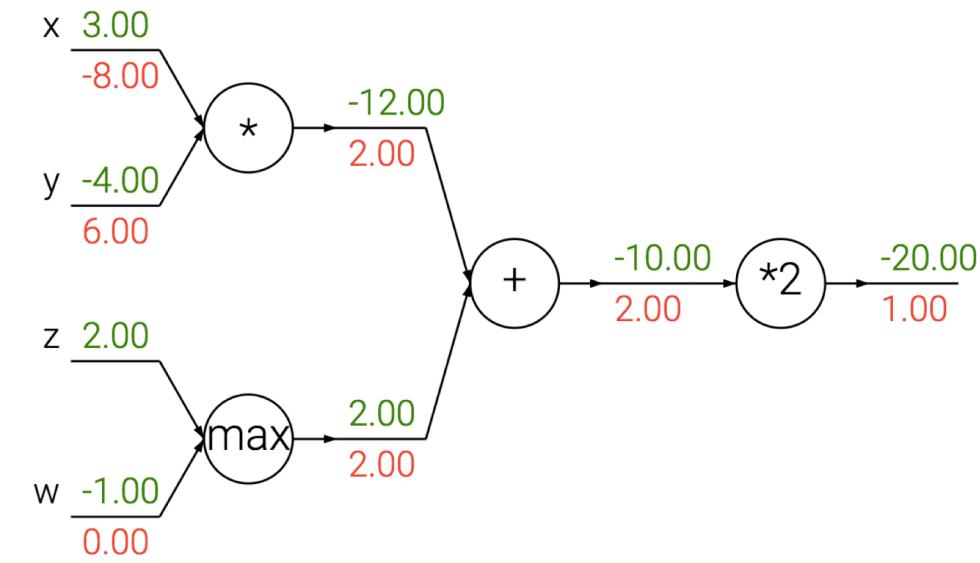


Add gate: “gradient distributor”

Multiply gate: “gradient switcher”

Source: [Stanford 231n](#)

Patterns in gradient flow



Add gate: “gradient distributor”

Multiply gate: “gradient switcher”

Max gate: “gradient router”

Source: [Stanford 231n](#)

Dealing with vectors

$$\frac{\partial z}{\partial x} = \begin{matrix} N \times M \\ \text{Jacobian} \end{matrix} \begin{pmatrix} \frac{\partial z_1}{\partial x_1} & \cdots & \frac{\partial z_1}{\partial x_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial z_N}{\partial x_1} & \cdots & \frac{\partial z_N}{\partial x_M} \end{pmatrix}$$

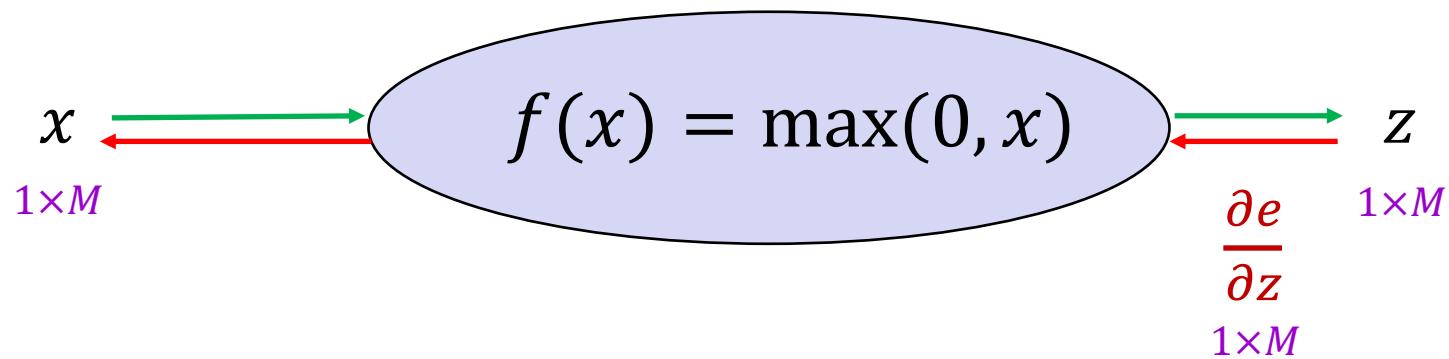
The diagram illustrates a function $f(x)$ represented by a light blue oval. Two green arrows point into the oval from the left, labeled x above and below. Two red arrows point out from the oval to the right, labeled z above and below. Below the input arrow x , the dimensions $1 \times M$ are given. Below the output arrow z , the dimensions $1 \times N$ are given. To the left of the input arrow, the expression $\frac{\partial e}{\partial x}$ is shown with dimensions $1 \times M$ above and $1 \times N$ below. To the right of the output arrow, the expression $\frac{\partial e}{\partial z}$ is shown with dimensions $N \times M$ above and $1 \times N$ below.

Simple case: Elementwise operation

Simple case: Elementwise operation (ReLU layer)

$$\frac{\partial z}{\partial x} = \begin{pmatrix} \frac{\partial z_1}{\partial x_1} & \cdots & \frac{\partial z_1}{\partial x_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial z_M}{\partial x_1} & \cdots & \frac{\partial z_M}{\partial x_M} \end{pmatrix}$$

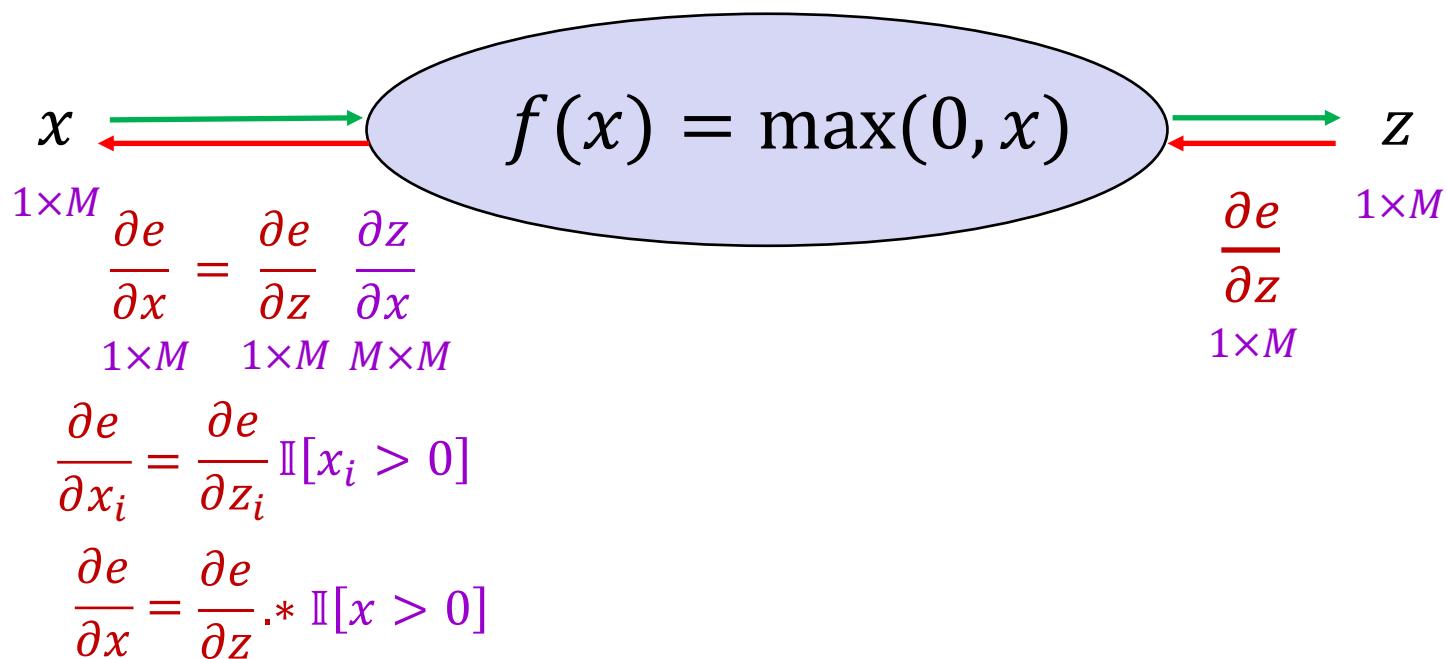
M × M
Jacobian



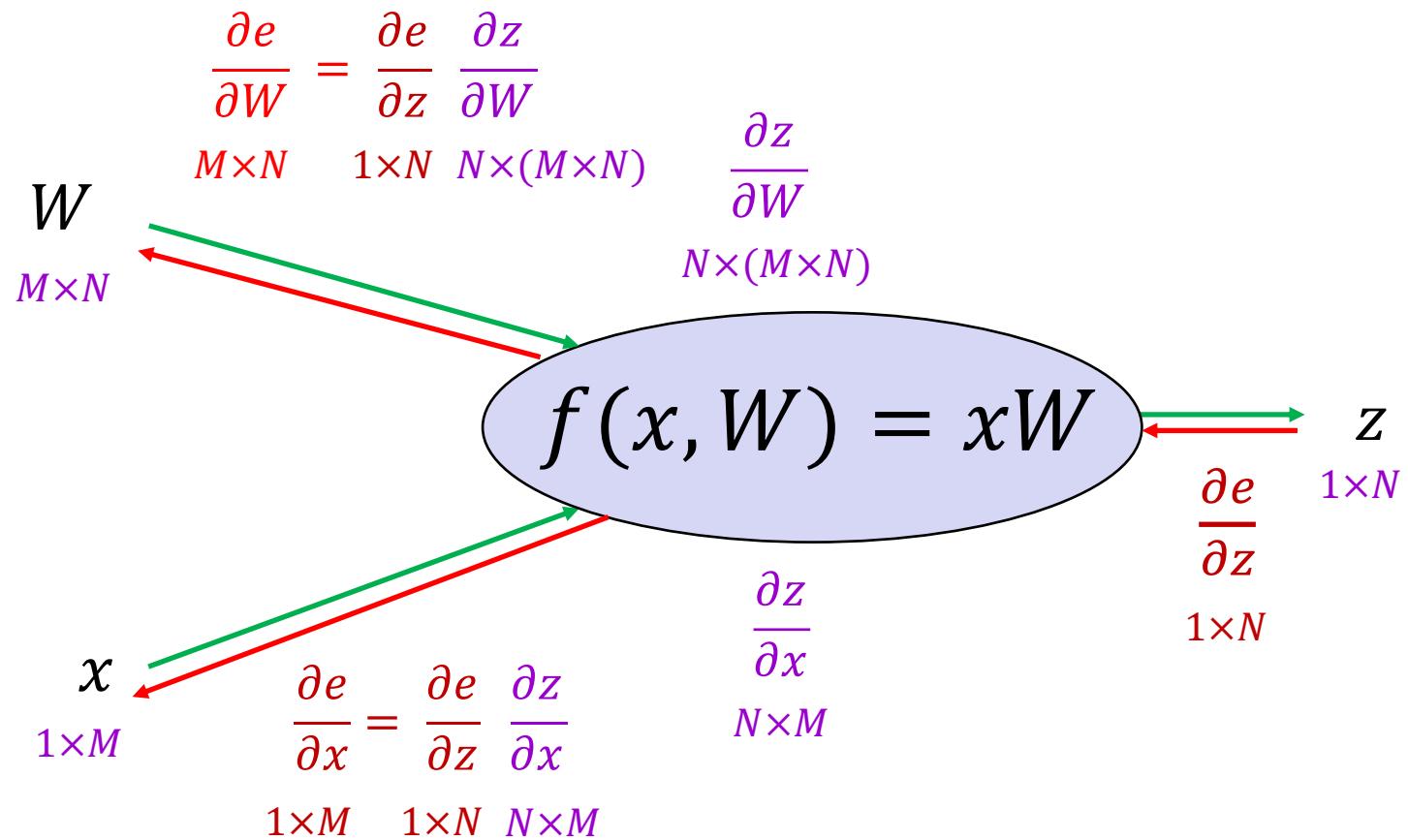
Simple case: Elementwise operation (ReLU layer)

$$\frac{\partial z}{\partial x} = \begin{pmatrix} \mathbb{I}[x_1 > 0] & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \mathbb{I}[x_M > 0] \end{pmatrix}$$

M × M
Jacobian



Matrix-vector multiplication (linear layer)



Matrix-vector multiplication (linear layer)

$$(z_1 \dots z_N) = (x_1 \dots x_M) \begin{pmatrix} W_{11} & \dots & W_{1N} \\ \vdots & \ddots & \vdots \\ W_{M1} & \dots & W_{MN} \end{pmatrix} \quad z_j = \sum_{i=1}^M x_i W_{ij}$$

Want: $\frac{\partial e}{\partial x} = \frac{\partial e}{\partial z} \boxed{\frac{\partial z}{\partial x}}$

$1 \times M \quad 1 \times N \quad N \times M$

$\frac{\partial z_j}{\partial x_i} =$ j th row, i th column
of Jacobian

$$\frac{\partial z}{\partial x} = W^T$$

$$\frac{\partial e}{\partial x} = \frac{\partial e}{\partial z} \frac{\partial z}{\partial x} = \frac{\partial e}{\partial z} W^T$$

Matrix-vector multiplication (linear layer)

$$(z_1 \dots z_N) = (x_1 \dots x_M) \begin{pmatrix} W_{11} & \dots & W_{1N} \\ \vdots & \ddots & \vdots \\ W_{M1} & \dots & W_{MN} \end{pmatrix} \quad z_j = \sum_{i=1}^M x_i W_{ij}$$

Want: $\frac{\partial e}{\partial W} = \frac{\partial e}{\partial z} \begin{bmatrix} \frac{\partial z}{\partial W} \end{bmatrix}$

$M \times N$ $1 \times N$ $N \times (M \times N)$

$$\frac{\partial z_k}{\partial W_{ij}} =$$

z_k depends only on
 k th column of W

$$\frac{\partial e}{\partial W_{ij}} =$$

$$\frac{\partial e}{\partial W} = x^T \frac{\partial e}{\partial z}$$

General tips

- Derive error signal (upstream gradient) directly, avoid explicit computation of huge local derivatives
 - Write out expression for a single element of the Jacobian, then deduce the overall formula
 - Keep consistent indexing conventions, order of operations
 - Use dimension analysis
-
- **For further reading:**
 - Lecture 4 of [Stanford 231n](#) and associated links in the syllabus
 - [Yes you should understand backprop](#) by Andrej Karpathy