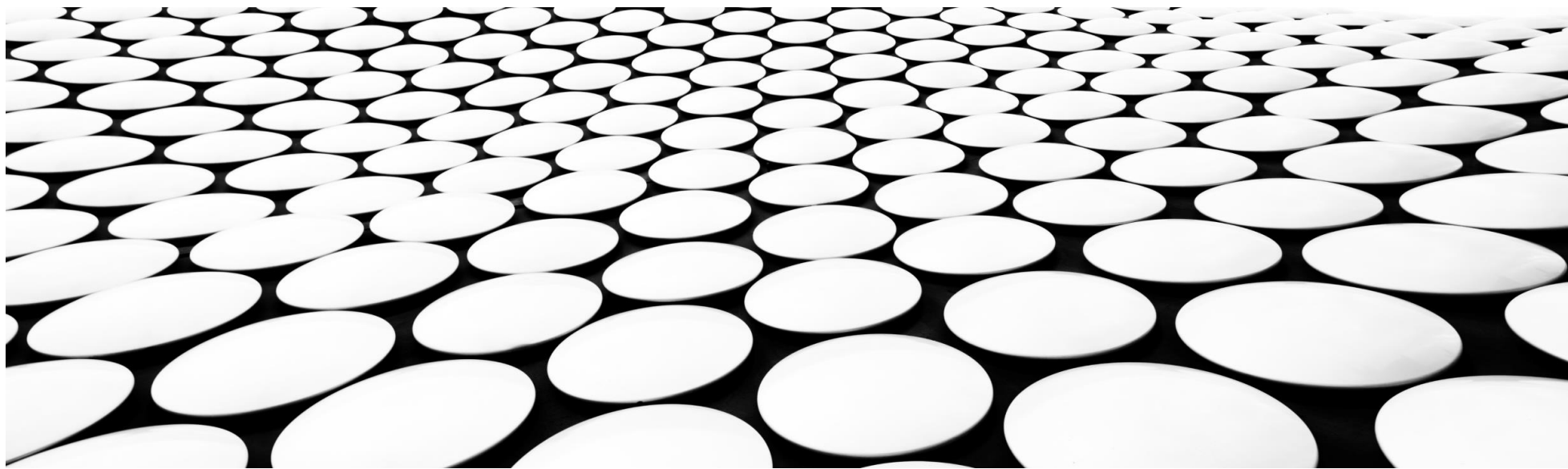


深度学习

邱怡轩



今天的主题

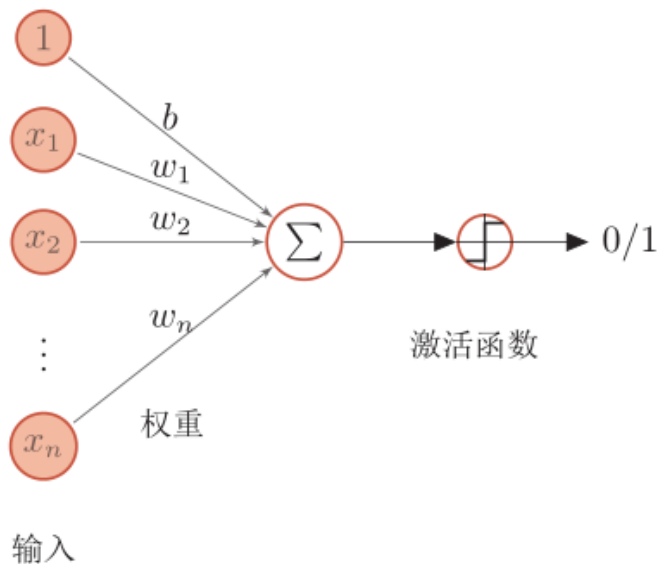
- 前馈神经网络
- 反向传播算法

前馈神经网络

前馈神经网络 \subset 人工神经网络

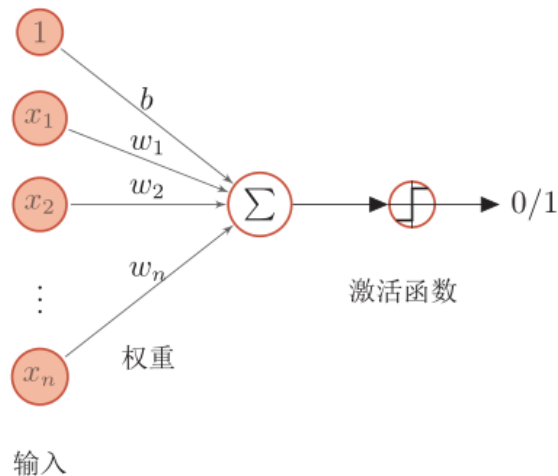
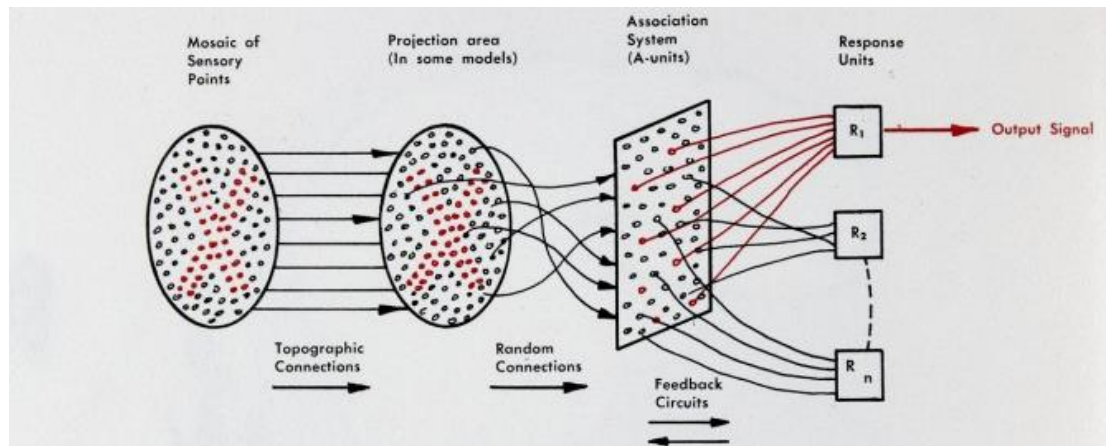
人工神经元

- 神经网络是连接主义模型的典型代表
- 基本构成单元为人工神经元
- “知识”存储在神经元之间的连接上
- 神经元负责信号/数据的输入和输出



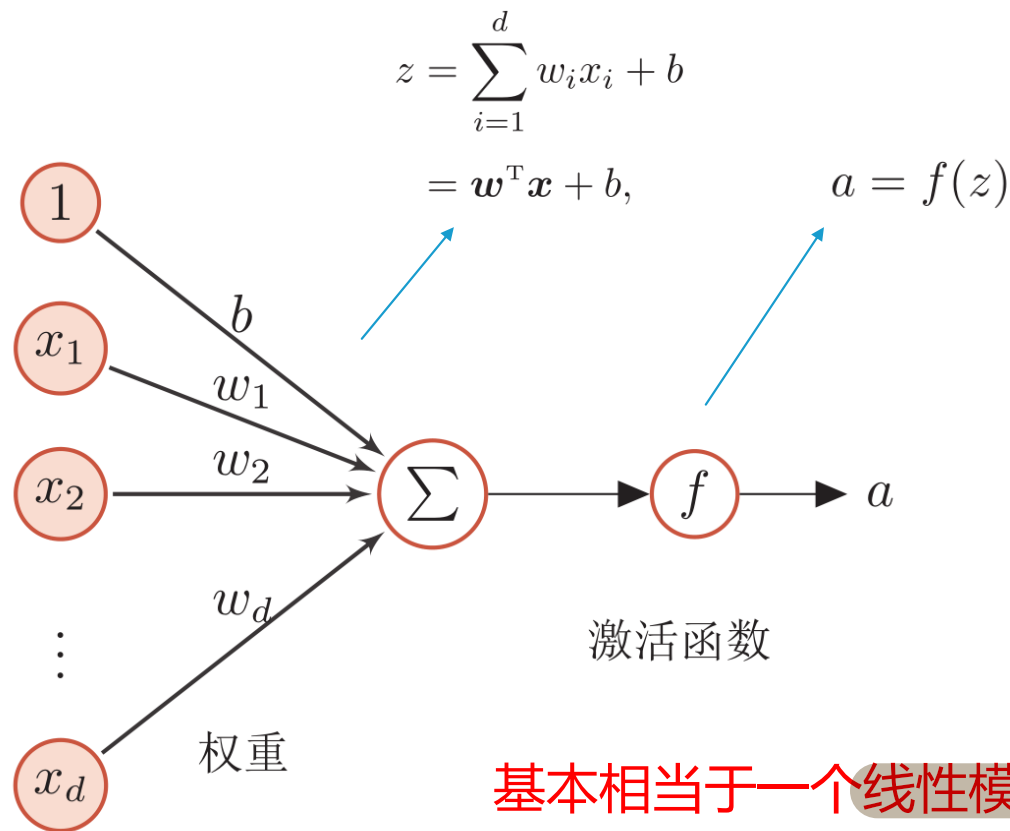
人工神经元

- 现代的人工神经元结构基本源自Rosenblatt



人工神经元

- 输入信号 → 加权求和 → 激活输出



激活函数

- Rosenblatt当时的激活函数是一个阶梯函数，

$$\text{例如 } a = f(z) = \begin{cases} 1, & z \geq 0.5 \\ 0, & z < 0.5 \end{cases}$$

- 当今广泛使用的激活函数一般满足如下性质：
 - 非线性
 - 连续，除了少数点外处处可导
 - 计算简单、数值稳定

常见 激活函数

- Sigmoid

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

映射到 $(0, 1)$ 之间
梯度消失 梯度爆炸

- Tanh

$$\tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}$$

解决 Sigmoid 不是 0 中心
梯度消失
 $(-1, 1)$

- ReLU

$$\text{ReLU}(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}$$
$$= \max(0, x).$$

简单, 收敛较快
并不是全区间可导
梯度不消失 不是 0 中心
 $[0, \infty)$

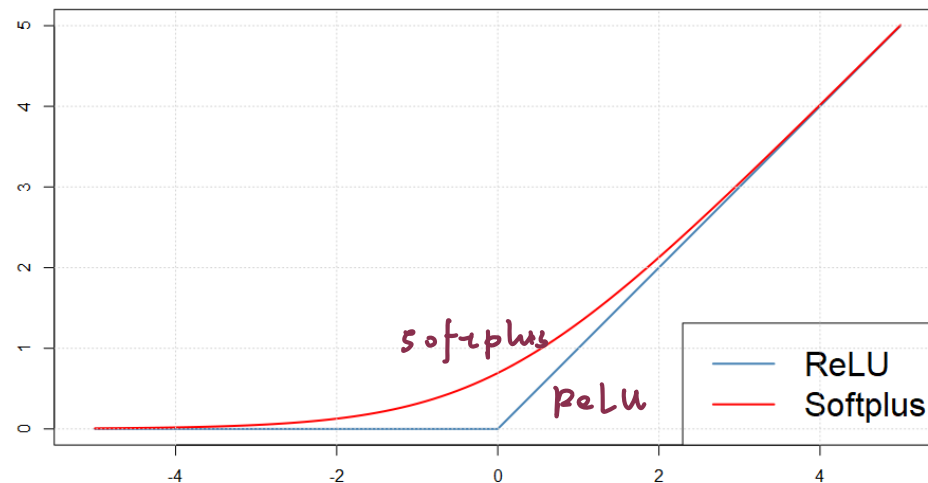
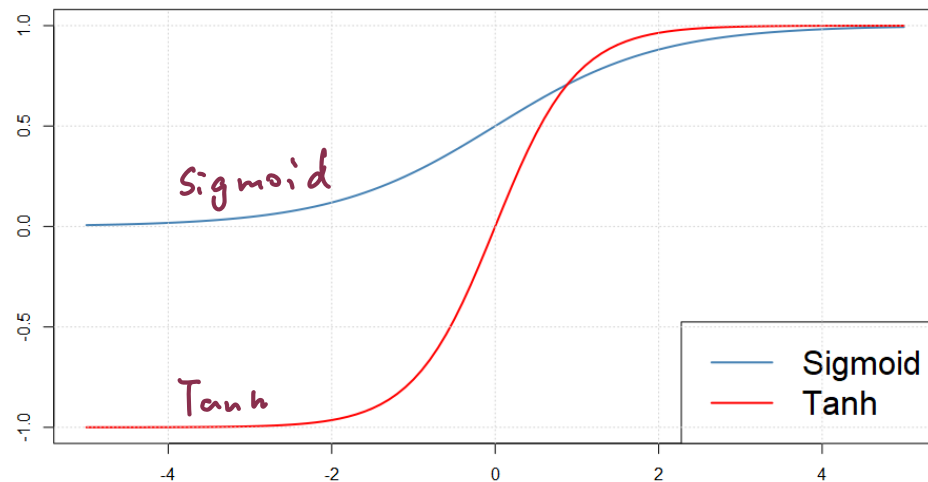
- Softplus

$$\text{softplus}(x) = \log(1 + \exp(x))$$

1/2-化 $(0, 1)$
多分类
 $(0, \infty)$

常见 激活函数

- Sigmoid
- Tanh
- ReLU
- Softplus



数值稳定

第一类

$$\text{sigmoid}(x) = \frac{1}{1+e^{-x}}$$

$$x \geq 0$$

$$\frac{1}{1+e^{-x}}$$

$$x < 0$$

$$\frac{e^x}{1+e^x}$$

先变绝对值

\rightarrow

$$\frac{1}{1+e}$$

$$\frac{e}{1+e}$$

torch.sigmoid

$$\text{Tanh}(x)$$

$$x \geq 0$$

$$\frac{1-e^{-2x}}{1+e^{-2x}}$$

$$x < 0$$

$$\frac{e^{2x}-1}{1+e^{2x}}$$

$$\frac{1-e}{1+e}$$

$$\frac{e-1}{1+e}$$

torch.tanh

第二类

$$\text{softplus}(x)$$

$$x \geq 0$$

$$x + \log(1+e^{-x})$$

$$x + \log$$

torch.nn.functional.softplus

$$\log(1+e^x)$$

$$x < 0$$

$$\log(1+e^x)$$

$$\log$$

$$\log(e^x + e^y)$$

$$x + \log(1+e^{y-x})$$

$$x > y$$

$$y + \log(1+e^{x-y})$$

$$x \leq y$$

数值稳定算法

- 在 Logistic 回归中，我们需要计算

sigmoid $\rho(x) = \frac{1}{1 + e^{-x}}$

- 为了计算目标函数还需要 $\log \rho(x)$ 和 $\log(1 - \rho(x))$

不能太接近 0.1
(log 似然中)

反正 $\rho(x)$ 为这个

$$\text{softmax} \quad \rho(x) = \frac{e^{x_i}}{e^{x_1} + e^{x_2} + \dots + e^{x_n}}$$
$$= \frac{1}{1 + e^{x_1 - x_i} + e^{x_2 - x_i} + \dots + e^{x_n - x_i}}$$

数值稳定算法

$$\log \rho(x) =$$

$$\rho(x) = \frac{1}{1+e^{-x}}$$

$$x - \log(1+e^x)$$

- **问题1**: 当 x 很小的负值时, $e^{-x} \rightarrow +\infty$, 造成 $\rho(x)$ 计算不稳定 $x = 100$
- **问题2**: 当 $\rho(x)$ 接近于0或1时, $\log \rho(x)$ 或 $\log(1 - \rho(x))$ 会出现 NaN

问题1

- 对于问题1, 一种解决方法是对 x 的取值
分类讨论

$$x \geq 0, \text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \leq 1$$

$$x < 0, \text{sigmoid}(x) = \frac{e^x}{1 + e^x} \leq 2$$

- 此时对于任意的 x , 分子与分母都是稳定的取值

`def sigmoid`

`if x >= 0: x 要尽可能同量化`

问题1

- `scipy.special.expit()` 函数进行了这样的处理
- 也可以很方便地手动实现

```
In [4]: import numpy as np
def sigmoid(x):
    x = np.array(x)
    e = np.exp(-np.abs(x))
    number = np.where(x >= 0.0, 1.0, e)
    denom = 1.0 + e
    return number / denom
```

不想重复计算
向量版 if else
向量

```
sigmoid([-1000, -100, -10, 0, 10, 100, 1000])
```

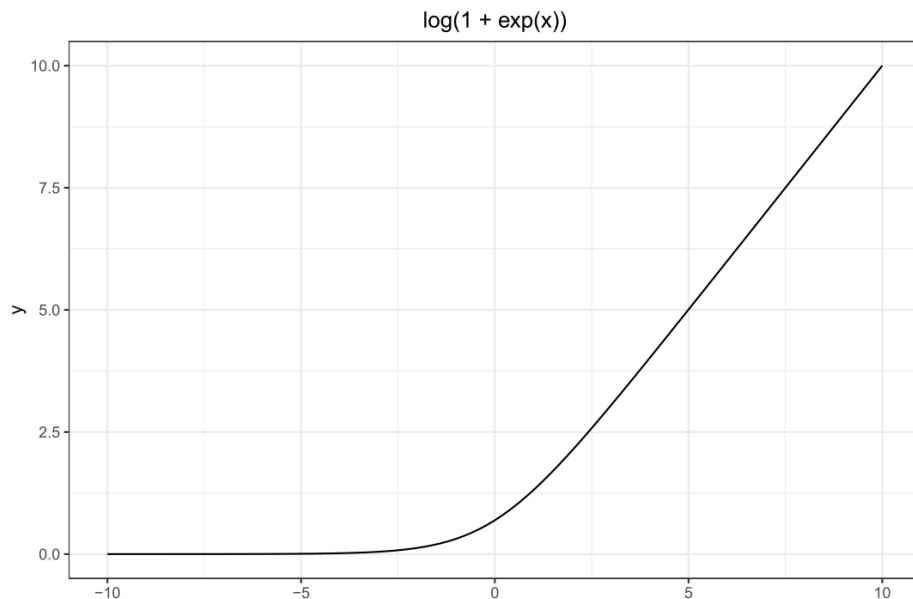
```
Out[4]: array([0.00000000e+00, 3.72007598e-44, 4.53978687e-05, 5.00000000e-01,
9.99954602e-01, 1.00000000e+00, 1.00000000e+00])
```

问题2

- 对于问题2, 一种简单粗暴的方法是在 $\log()$ 函数中加上一个很小的正数
- 但还有一种更好的方式

问题2

- 事实上, $s(x) = \log(1 + e^x)$ 是一个数值稳定的函数
- 但需要用特殊的计算方法



问题2

- 如果直接计算, 那么 x 很大时 $\exp(x)$ 将会溢出
- 但是可以发现
$$s(x) = \log(1 + e^x) = x + \log(1 + e^{-x})$$
在 x 很大时与 x 是同一量级
- 因此可以分类讨论

$$s(x) = \begin{cases} \log(1 + e^x), & x < 0 \\ x + \log(1 + e^{-x}), & x \geq 0 \end{cases}$$

- 思考: 如何用 Numpy 进行向量化实现

$$\sum y_i \log p_i + (1 - y_i) \log (1 - p_i)$$

↑
 $s(x)$ 替代

$$s(x) = 1 + e^x$$

1 1 e 1 0

不知道怎么选择时

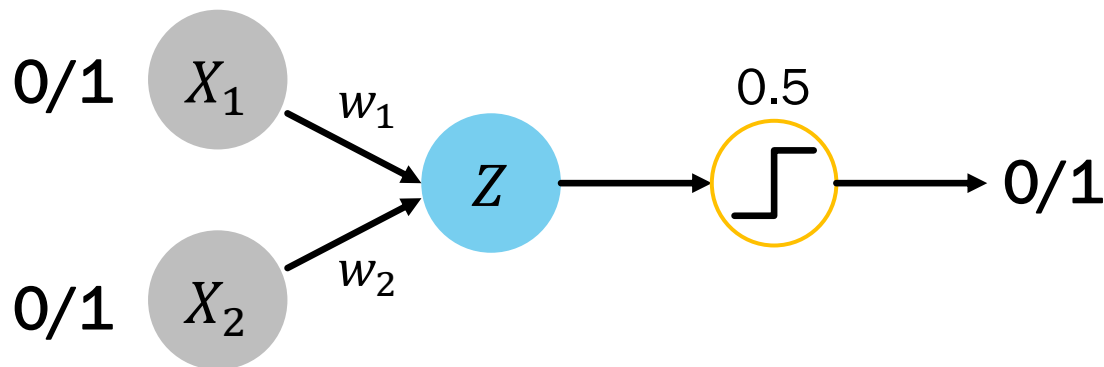
就以ReLU作为默认选项

编程实现

- 在编程实现中需要特别注意数值稳定性
- 特别是牵涉到指数函数 $\exp(x)$

思考题

- 试着用Rosenblatt的感知器模拟逻辑运算
- 两个输入: $X_1 \in \{0,1\}$, $X_2 \in \{0,1\}$
- 线性函数: $z = w_1X_1 + w_2X_2$
- 一个输出: 若 $z > 0.5$, 则 $Y = 1$, 否则 $Y = 0$



思考题

- 尝试找出适当的参数 w_1 和 w_2 ，使得这个人工神经元可以实现

1. AND运算
2. OR运算
3. XOR运算

INPUT		OUTPUT
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

INPUT		OUTPUT
A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

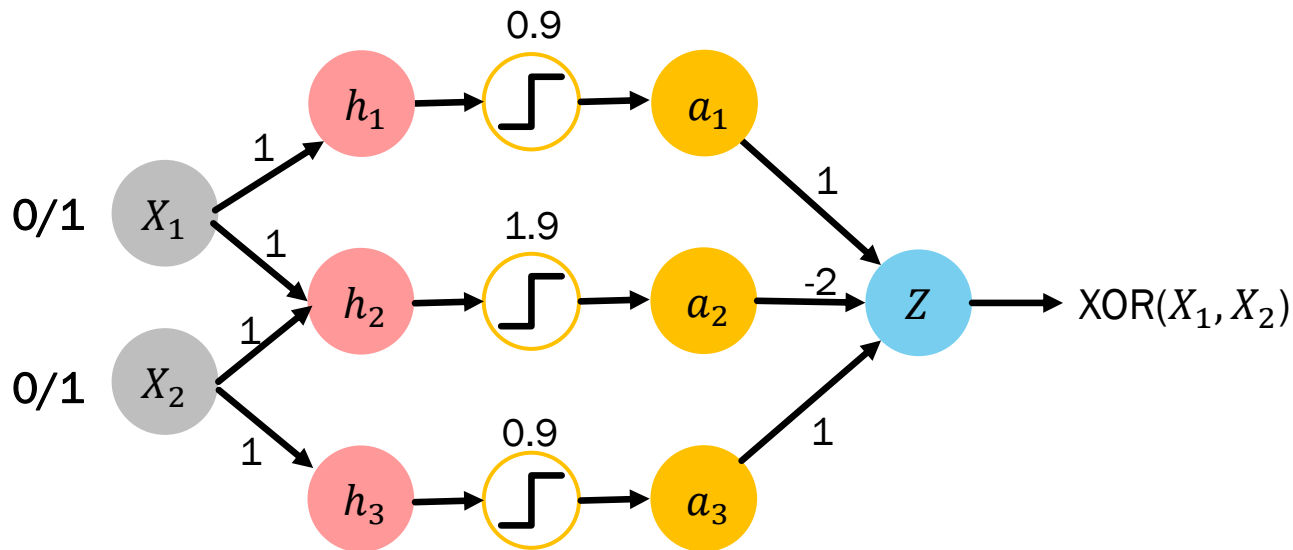
INPUT		OUTPUT
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

启示

- 单层的感知器具有很大的局限
- 可以将神经元串联起来
- 构建多层的神经网络

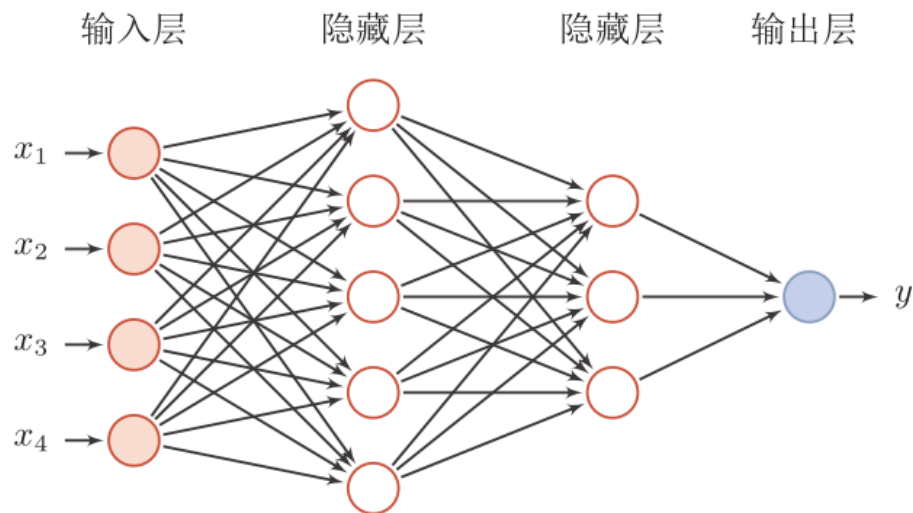
XOR问题

- 只需增加一层神经元，即可模拟XOR运算



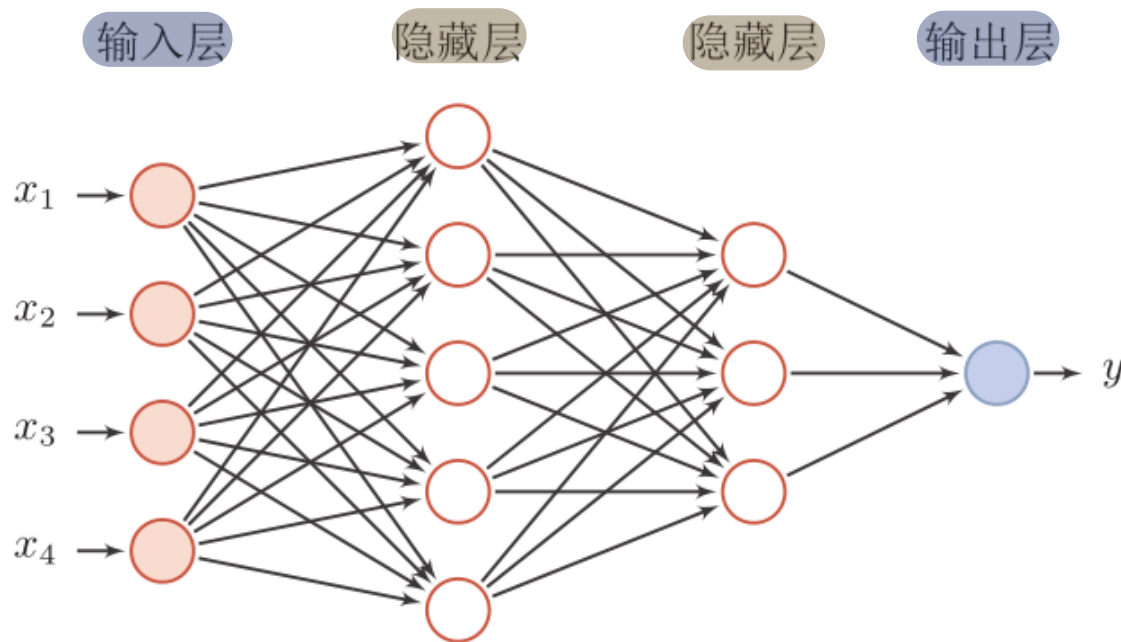
前馈神经网络

- 将神经元按照一定的规则组合，可以得到复杂的神经网络
- 前馈神经网络是一种结构相对简单的神经网络
- 也称为全连接神经网络、多层感知器



前馈神经网络

- 各神经元分别属于不同的层，**层内无连接**
- **相邻两层**之间的神经元**全部两两连接**
- 信号从输入层向输出层**单向传播**



正式定义

本页内容取自邱锡鹏
《神经网络与深度学习》

记号	含义
L	神经网络的层数
M_l	第 l 层神经元的个数
$f_l(\cdot)$	第 l 层神经元的激活函数
$\mathbf{W}^{(l)} \in \mathbb{R}^{M_l \times M_{l-1}}$	第 $l-1$ 层到第 l 层的权重矩阵
$\mathbf{b}^{(l)} \in \mathbb{R}^{M_l}$	第 $l-1$ 层到第 l 层的偏置
$\mathbf{z}^{(l)} \in \mathbb{R}^{M_l}$	第 l 层神经元的净输入 (净活性值)
$\mathbf{a}^{(l)} \in \mathbb{R}^{M_l}$	第 l 层神经元的输出 (活性值)

输入 $\mathbf{z}^{(l)} = \mathbf{W}^{(l)} \mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}$,

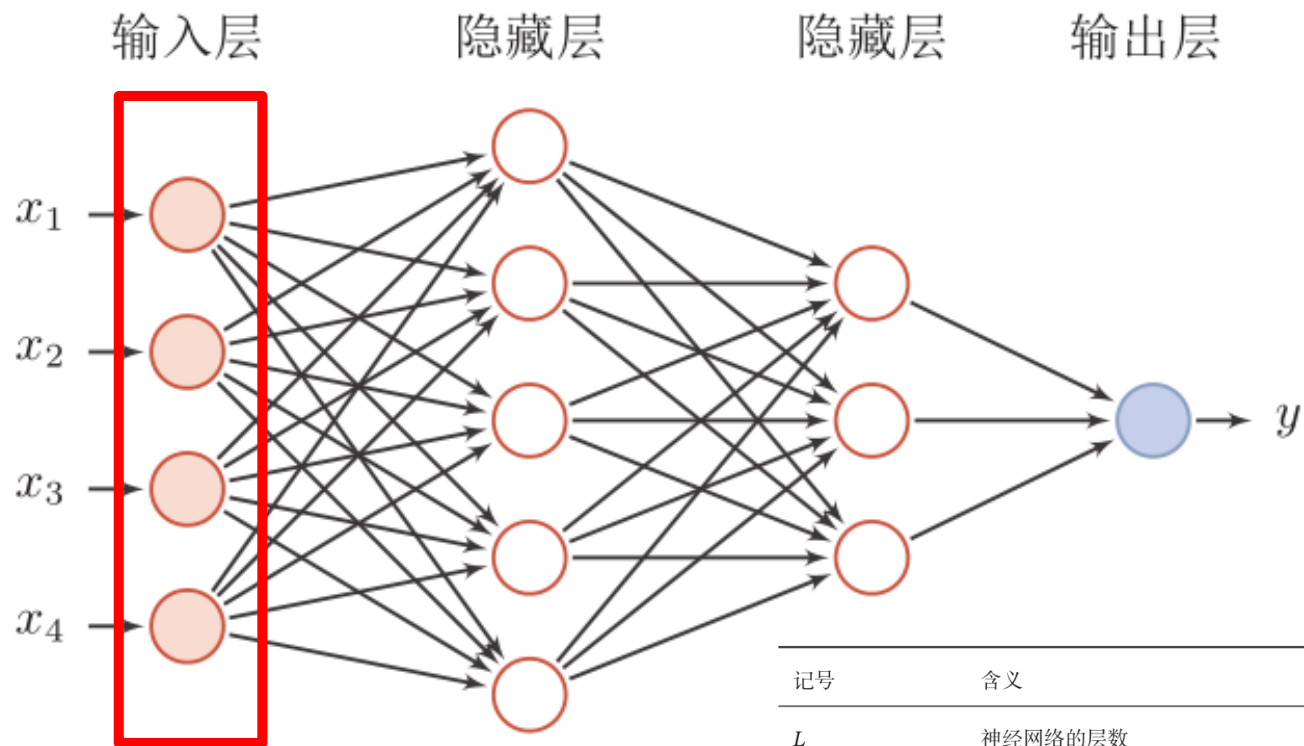
输出 $\mathbf{a}^{(l)} = f_l(\mathbf{z}^{(l)})$.

激活函数

■ 信息前馈传播方式

$$\mathbf{x} = \mathbf{a}^{(0)} \rightarrow \mathbf{z}^{(1)} \rightarrow \mathbf{a}^{(1)} \rightarrow \mathbf{z}^{(2)} \rightarrow \dots \rightarrow \mathbf{a}^{(L-1)} \rightarrow \mathbf{z}^{(L)} \rightarrow \mathbf{a}^{(L)} = \phi(\mathbf{x}; \mathbf{W}, \mathbf{b})$$

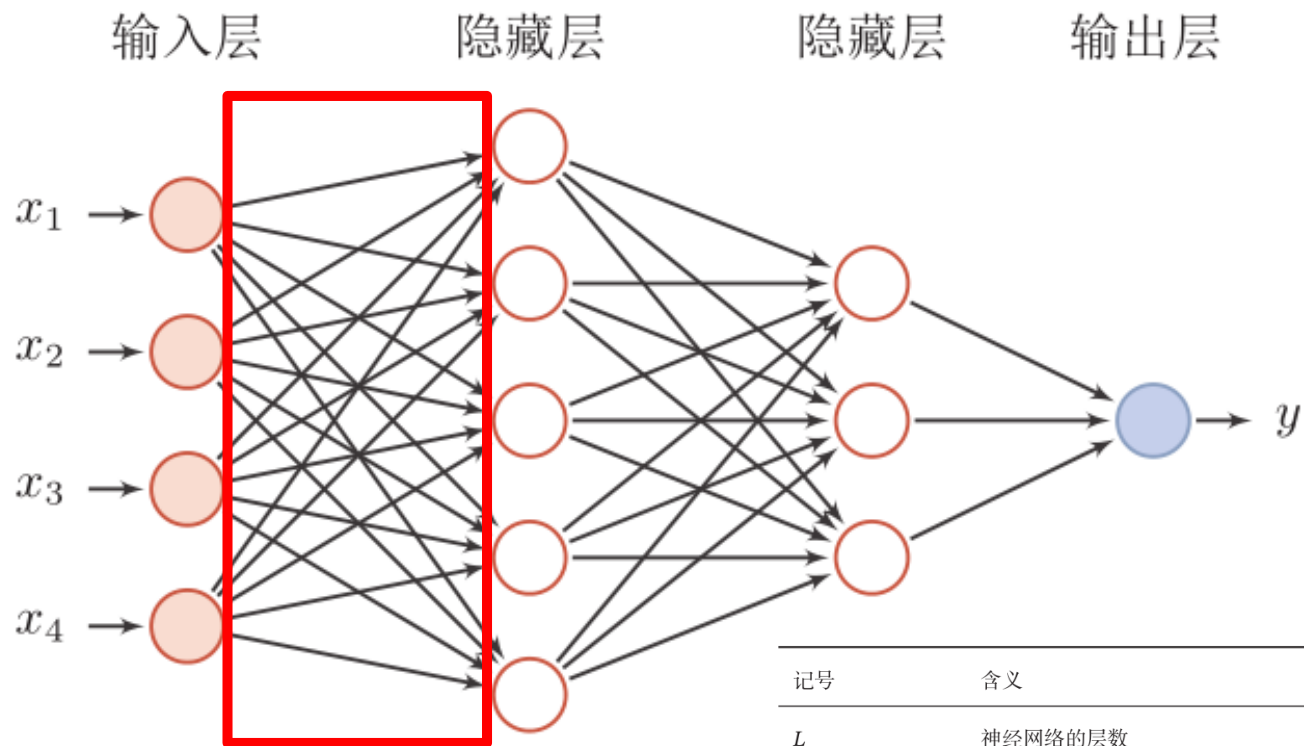
前馈神经网络



$$a^{(0)} = x \in \mathbb{R}^4$$

记号	含义
L	神经网络的层数
M_l	第 l 层神经元的个数
$f_l(\cdot)$	第 l 层神经元的激活函数
$\mathbf{w}^{(l)} \in \mathbb{R}^{M_l \times M_{l-1}}$	第 $l-1$ 层到第 l 层的权重矩阵
$\mathbf{b}^{(l)} \in \mathbb{R}^{M_l}$	第 $l-1$ 层到第 l 层的偏置
$\mathbf{z}^{(l)} \in \mathbb{R}^{M_l}$	第 l 层神经元的净输入 (净活性值)
$\mathbf{a}^{(l)} \in \mathbb{R}^{M_l}$	第 l 层神经元的输出 (活性值)

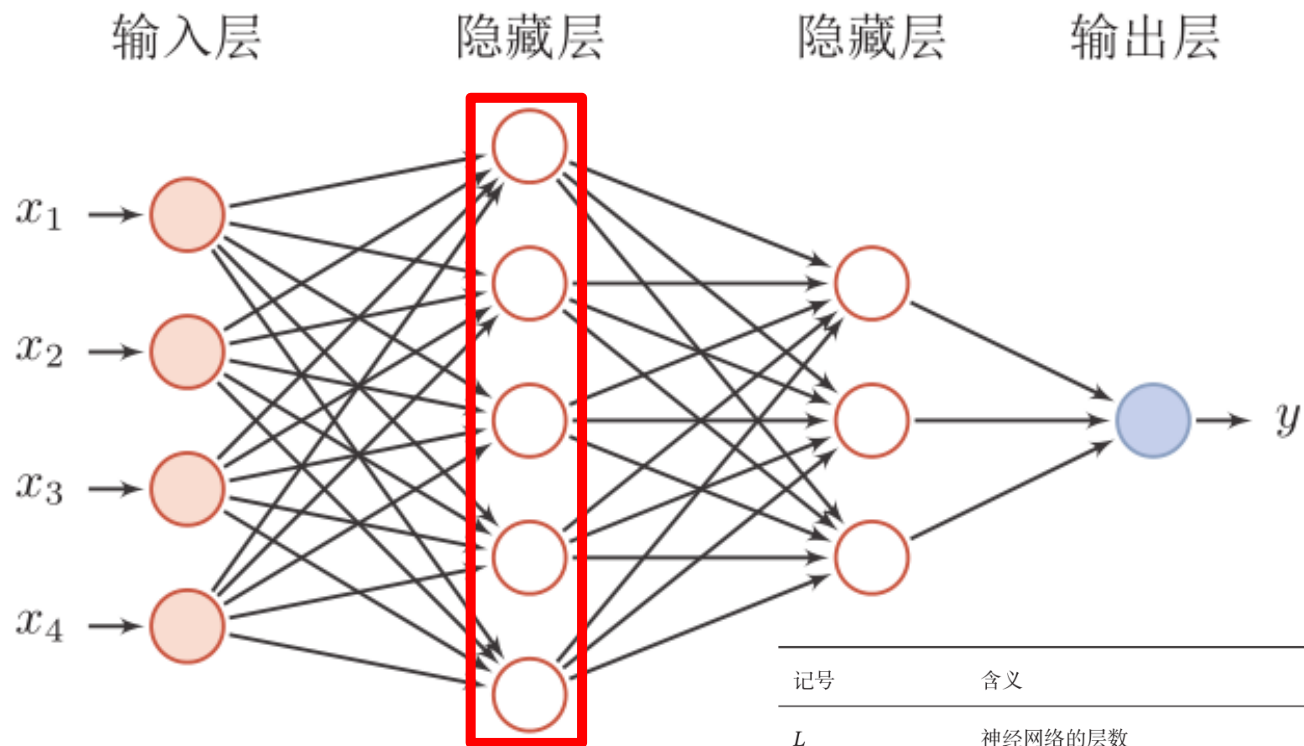
前馈神经网络



$$W^{(1)} \in \mathbb{R}^{5 \times 4}$$

记号	含义
L	神经网络的层数
M_l	第 l 层神经元的个数
$f_l(\cdot)$	第 l 层神经元的激活函数
$W^{(l)} \in \mathbb{R}^{M_l \times M_{l-1}}$	第 $l-1$ 层到第 l 层的权重矩阵
$b^{(l)} \in \mathbb{R}^{M_l}$	第 $l-1$ 层到第 l 层的偏置
$z^{(l)} \in \mathbb{R}^{M_l}$	第 l 层神经元的净输入 (净活性值)
$a^{(l)} \in \mathbb{R}^{M_l}$	第 l 层神经元的输出 (活性值)

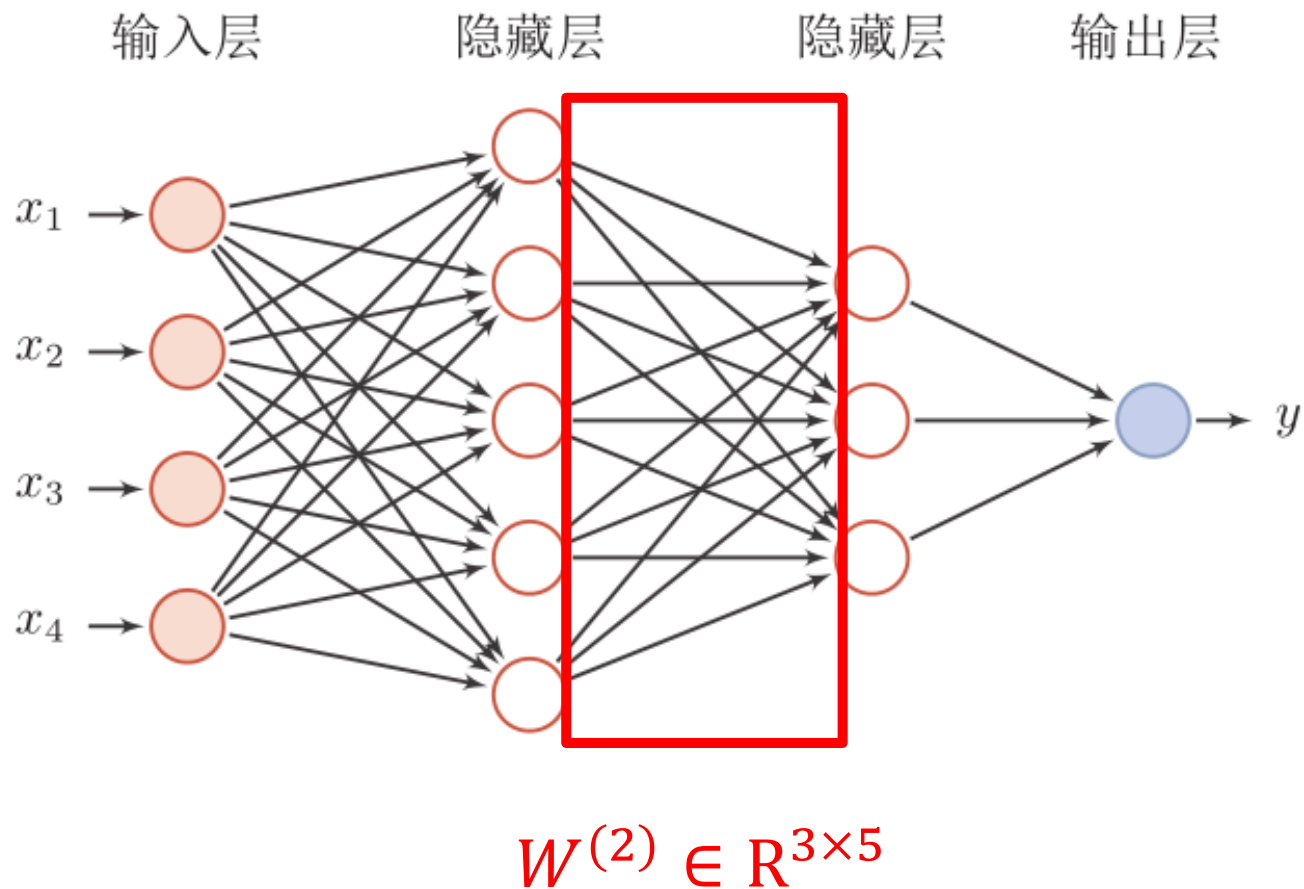
前馈神经网络



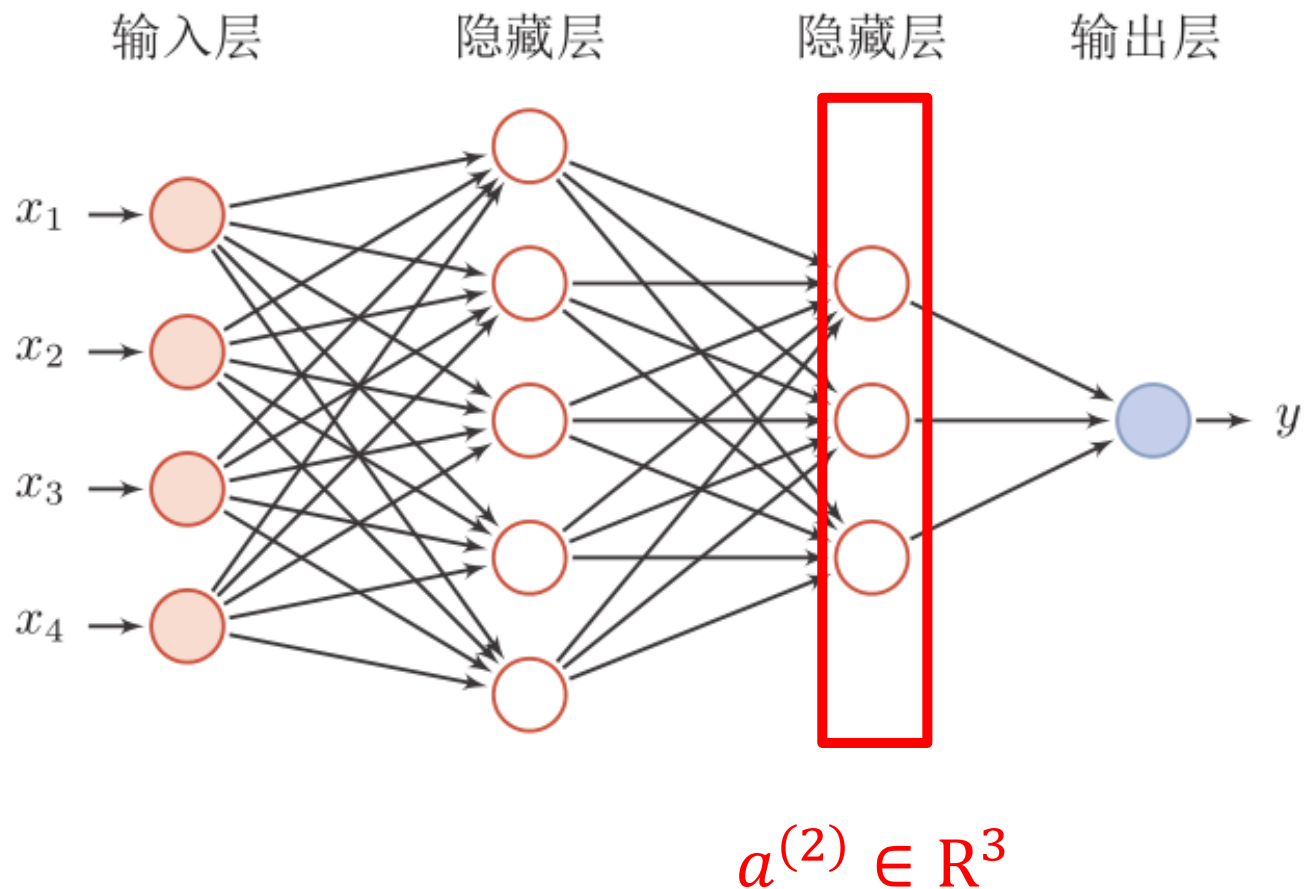
$$a^{(1)} \in \mathbb{R}^5$$

记号	含义
L	神经网络的层数
M_l	第 l 层神经元的个数
$f_l(\cdot)$	第 l 层神经元的激活函数
$\mathbf{W}^{(l)} \in \mathbb{R}^{M_l \times M_{l-1}}$	第 $l-1$ 层到第 l 层的权重矩阵
$\mathbf{b}^{(l)} \in \mathbb{R}^{M_l}$	第 $l-1$ 层到第 l 层的偏置
$\mathbf{z}^{(l)} \in \mathbb{R}^{M_l}$	第 l 层神经元的净输入 (净活性值)
$\mathbf{a}^{(l)} \in \mathbb{R}^{M_l}$	第 l 层神经元的输出 (活性值)

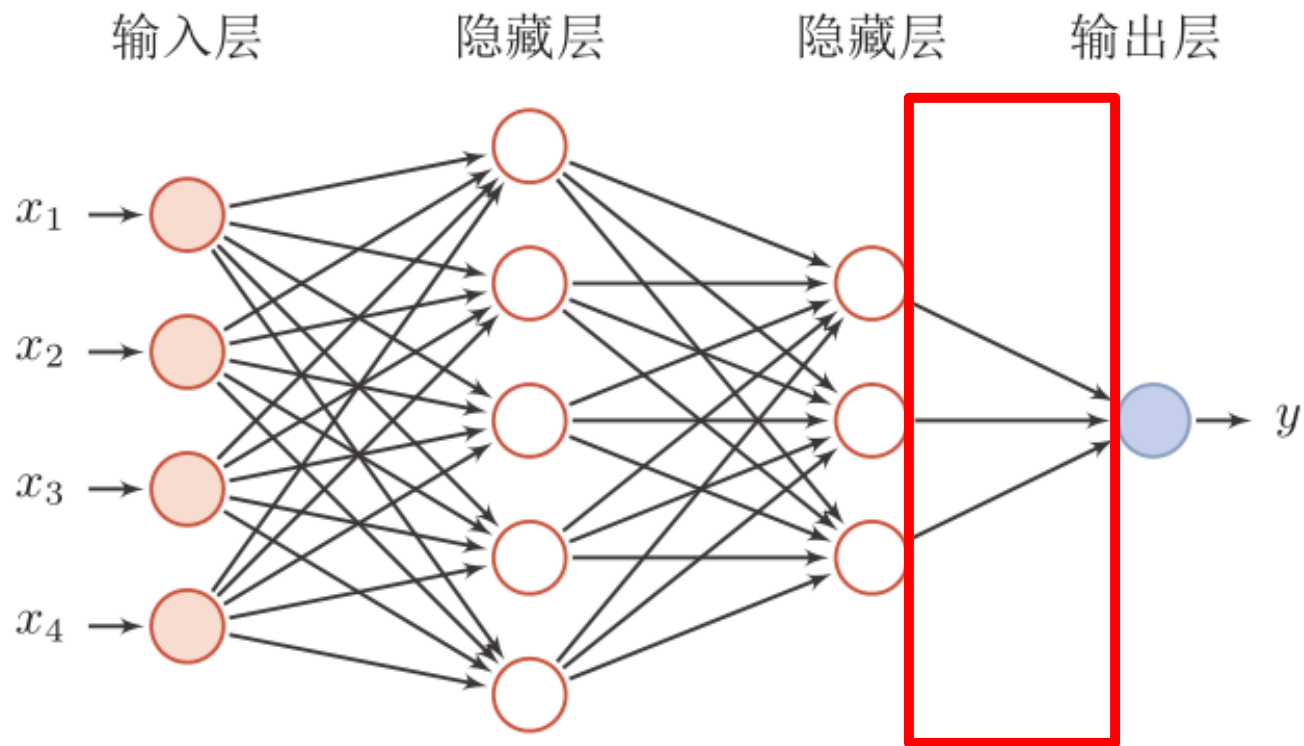
前馈神经网络



前馈神经网络

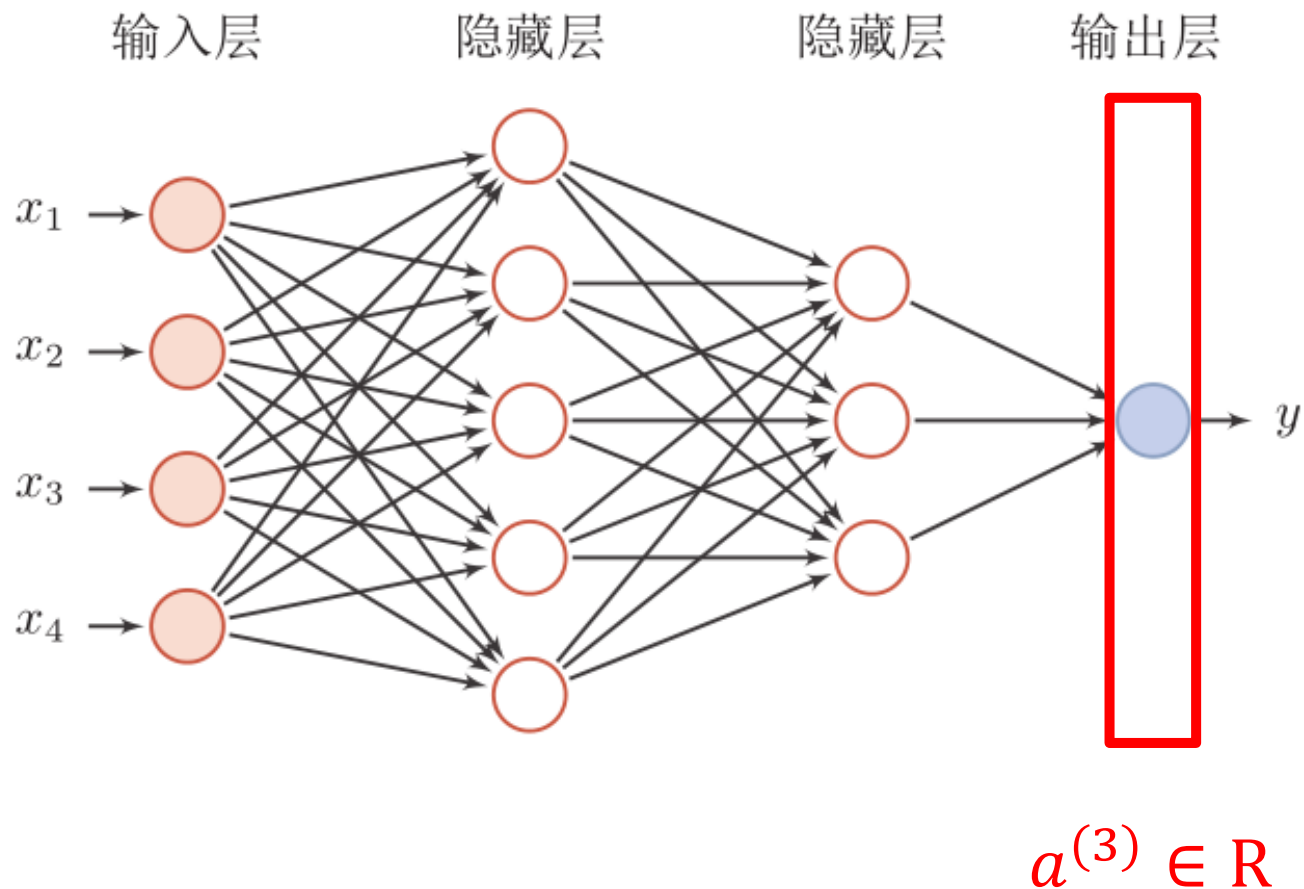


前馈神经网络



$$W^{(3)} \in \mathbb{R}^{1 \times 3}$$

前馈神经网络



前馈神经网络

- 本质上，前馈神经网络是一个复合函数
- $y = f^5(f^4(f^3(f^2(f^1(x))))))$
- 每一个 f^i 可以是固定的（如激活函数），或者是带参数的（如带权重的线性变换）
- 每一个 f^i 都相对简单
- 但复合之后可以变得非常复杂

通用近似定理

本页定理取自邱锡鹏
《神经网络与深度学习》

- 之前我们展示了，一个两层的前馈神经网络可以拟合XOR函数
- 事实上，**两层的网络几乎可以拟合任意函数！**

定理 4.1 – 通用近似定理 (Universal Approximation Theorem)

[Cybenko, 1989, Hornik et al., 1989]: 令 $\varphi(\cdot)$ 是一个非常数、有界、单调递增的连续函数, \mathcal{I}_d 是一个 d 维的单位超立方体 $[0, 1]^d$, $C(\mathcal{I}_d)$ 是定义在 \mathcal{I}_d 上的连续函数集合。对于任何一个函数 $f \in C(\mathcal{I}_d)$, 存在一个整数 m , 和一组实数 $v_i, b_i \in \mathbb{R}$ 以及实数向量 $\mathbf{w}_i \in \mathbb{R}^d$, $i = 1, \dots, m$, 以至于我们可以定义函数

$$F(\mathbf{x}) = \sum_{i=1}^m v_i \varphi(\mathbf{w}_i^T \mathbf{x} + b_i), \quad (4.33)$$

作为函数 f 的近似实现, 即

$$|F(\mathbf{x}) - f(\mathbf{x})| < \epsilon, \forall \mathbf{x} \in \mathcal{I}_d. \quad (4.34)$$

其中 $\epsilon > 0$ 是一个很小的正数。

题外话

- Kurt Hornik 何许人也?



题外话

- 其贡献在神经网络领域可以“封神”
- 网上几乎找不到关于他的其他照片
- 只有一个简陋的德语维基页面，无英文版
- 很少出现在公开学术场合

[Multilayer feedforward networks are universal approximators](#)

K Hornik, M Stinchcombe, H White - *Neural networks*, 1989 - Elsevier

This paper rigorously establishes that standard multilayer feedforward networks with as few as one hidden layer using arbitrary squashing functions are capable of approximating any Borel measurable function from one finite dimensional space to another to any desired ...

☆ 77 Cited by 20587 Related articles All 14 versions »

[Approximation capabilities of multilayer feedforward networks](#)

K Hornik - *Neural networks*, 1991 - Elsevier

We show that standard multilayer feedforward networks with as few as a single hidden layer and arbitrary bounded and nonconstant activation function are universal approximators with respect to $L_p(\mu)$ performance criteria, for arbitrary finite input environment measures μ ...

☆ 77 Cited by 5223 Related articles All 11 versions

[Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks](#)

K Hornik, M Stinchcombe, H White - *Neural networks*, 1990 - Elsevier

We give conditions ensuring that multilayer feedforward networks with as few as a single hidden layer and an appropriately smooth hidden layer activation function are capable of arbitrarily accurate approximation to an arbitrary function and its derivatives. In fact, these ...

☆ 77 Cited by 2059 Related articles All 8 versions

题外话

- 更为人知的一个身份是R的核心开发组成员
- CRAN的主要架构师和维护者
- e1071, kernlab, RWeka, arules, party, tm等重要R包的作者
- 其贡献不应该被埋没



Kurt Hornik <Kurt.Hornik@wu.ac.at>
to Kurt, Brian, Uwe ▾

Wed, Dec 11, 2013, 7:28 PM



Dear CRAN package maintainer,

We email all CRAN maintainers periodically to check that the email address is still active and to remind them of the CRAN policies at <http://CRAN.R-project.org/web/packages/policies.html>.

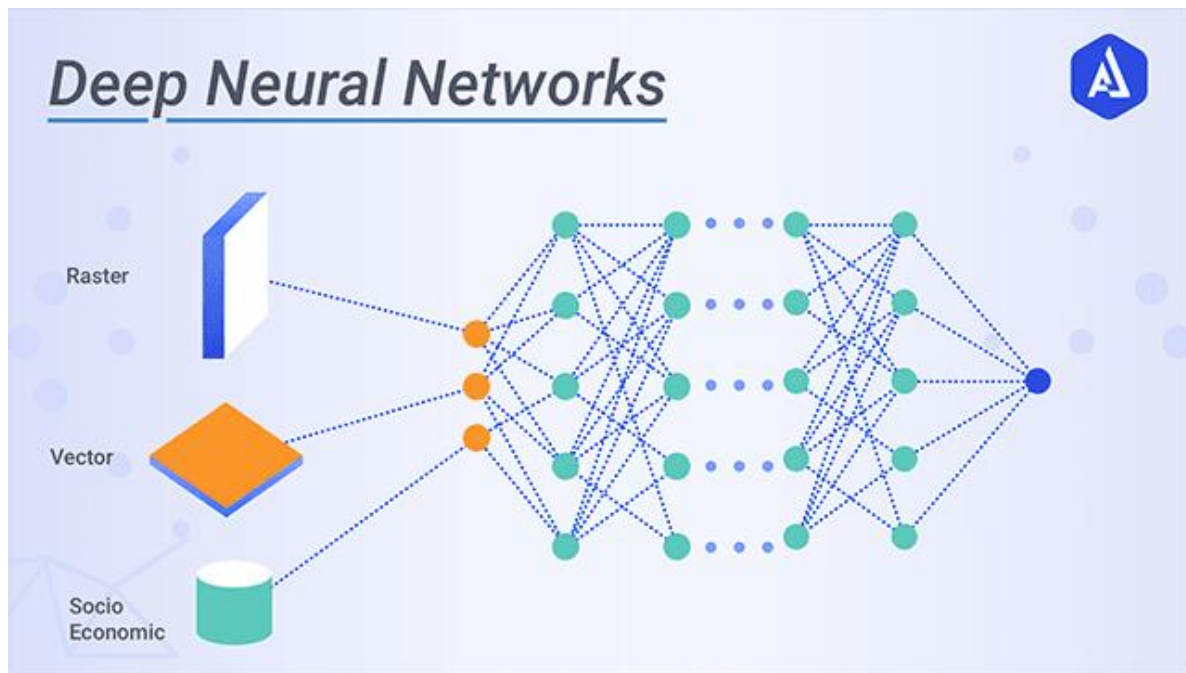
Changes to those policies in the last 4 months are given below.

Please check whether your maintainer address needs updating (in particular in case you moved, or received multiple copies of this message).

Best,
-k

深度神经网络

- 根据通用近似定理，两层的前馈神经网络已经可以近似绝大部分函数
- 但一些新的研究表明，网络的深度可能发挥着重要的作用



深度 神经网络

- 对于网络深度的研究是当前的热点课题

模型训练

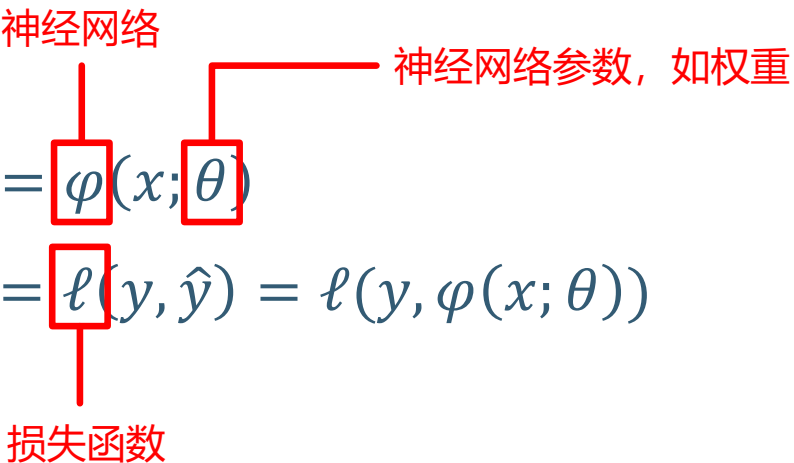
- 神经网络作为一个函数类具有良好的性质
- “万能函数”
- 接下来的问题在于如何估计其中的参数

神经网络

神经网络参数，如权重

- $\hat{y} = \varphi(x; \theta)$
- $L = \ell(y, \hat{y}) = \ell(y, \varphi(x; \theta))$

损失函数



梯度下降法

导数和梯度关系？

- $\theta_{k+1} = \theta_k - \frac{\partial \ell(y, \varphi(x; \theta))}{\partial \theta} \Big|_{\theta = \theta_k}$
- 核心在于求损失函数对参数的导数

反向传播算法

→ 自动微分

反向传播 算法

- 反向传播算法 (Backpropagation, BP)
- 并不神秘
- 本质上是一种聪明、高效地求导数的办法

一些吐槽

- BP的地位比较尴尬
- 首先它确实很重要，甚至一度推动了历史的进程
- 但如今几乎已经被更通用的自动微分取代
- 现实中你以99.9%的可能不需要自己实现

一些吐槽

- 为什么还要学习BP?
- 魔鬼在细节中
- 理解工作机制
- 指导如何调参
- 辅助Debugging
- 顺便学习了一些其他有用的知识



求导

- 求导就像搬砖
- 总是可以用基础的方法慢慢来
- 但有些方法可以更省力、更省时

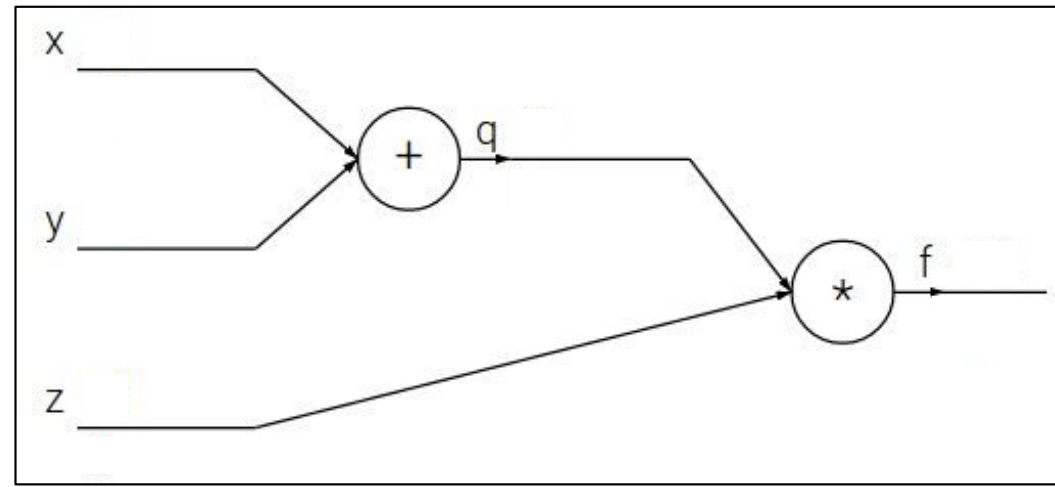
以下内容来自
<http://cs231n.stanford.edu>

Backpropagation: a simple example

$$f(x, y, z) = (x + y)z$$

Backpropagation: a simple example

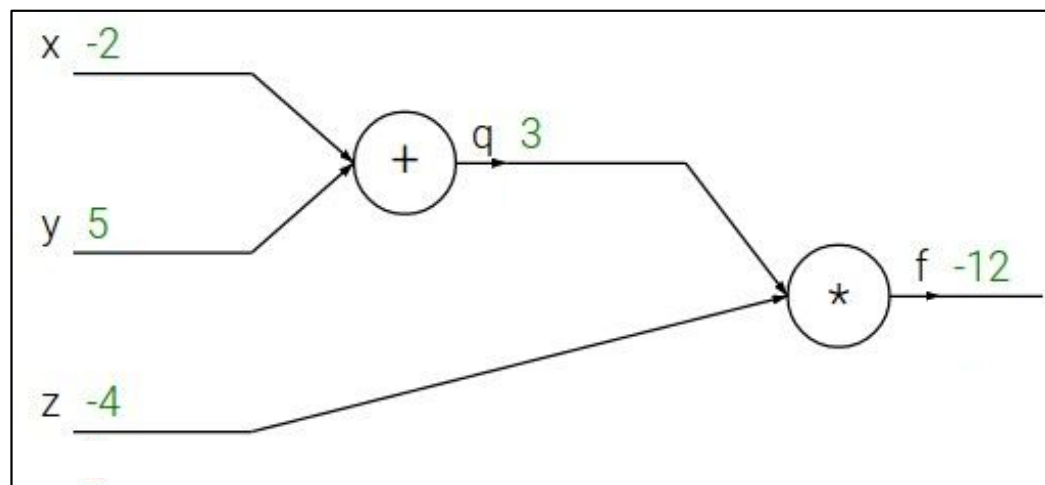
$$f(x, y, z) = (x + y)z$$



Backpropagation: a simple example

$$f(x, y, z) = (x + y)z$$

e.g. $x = -2$, $y = 5$, $z = -4$

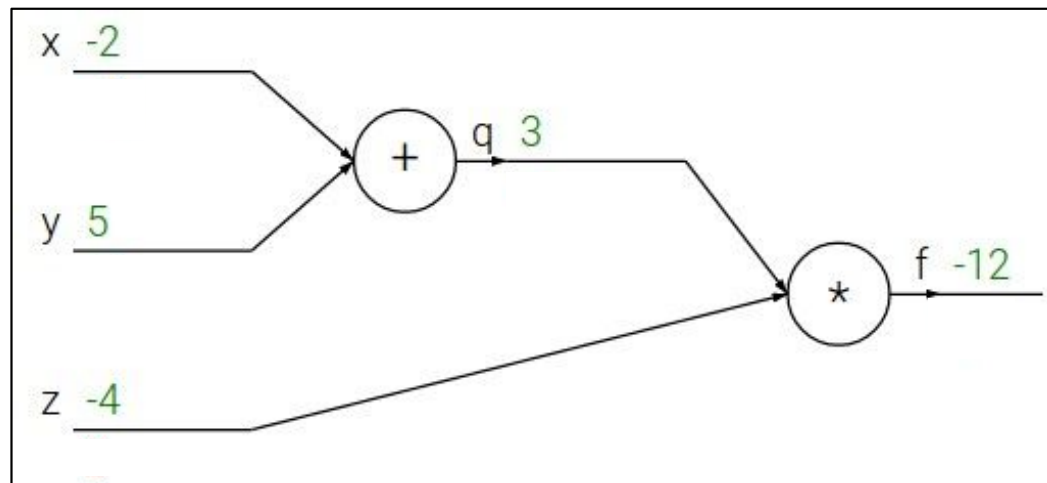


Backpropagation: a simple example

$$f(x, y, z) = (x + y)z$$

e.g. $x = -2$, $y = 5$, $z = -4$

$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$



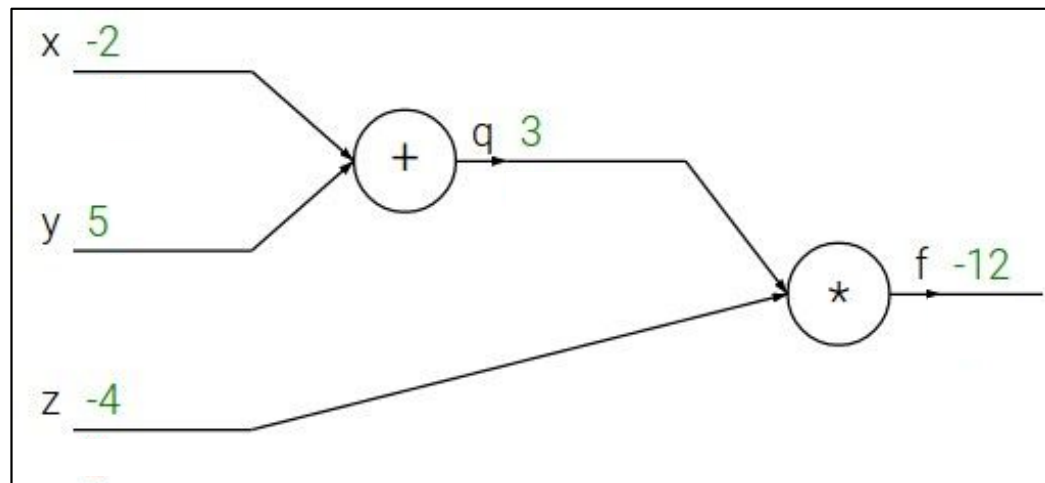
Backpropagation: a simple example

$$f(x, y, z) = (x + y)z$$

e.g. $x = -2$, $y = 5$, $z = -4$

$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \quad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$



Backpropagation: a simple example

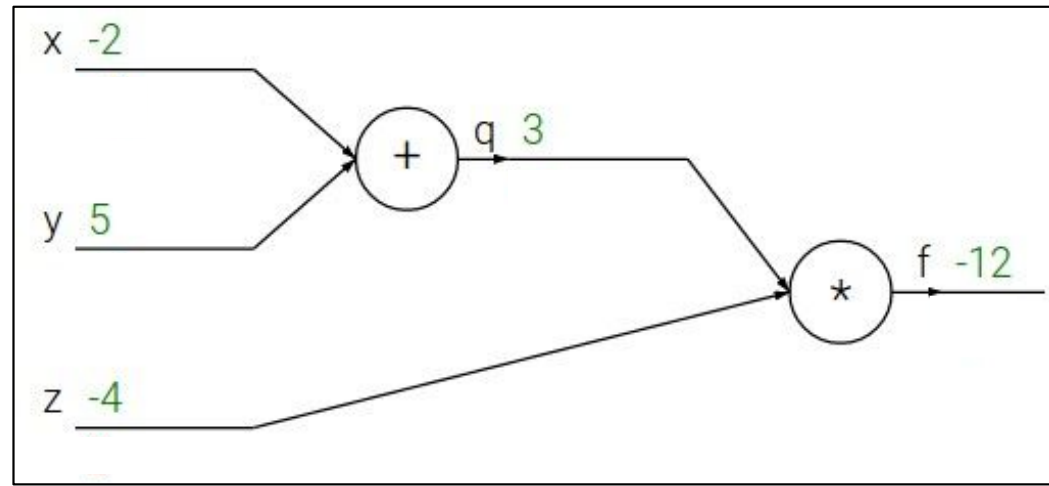
$$f(x, y, z) = (x + y)z$$

e.g. $x = -2$, $y = 5$, $z = -4$

$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \quad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



Backpropagation: a simple example

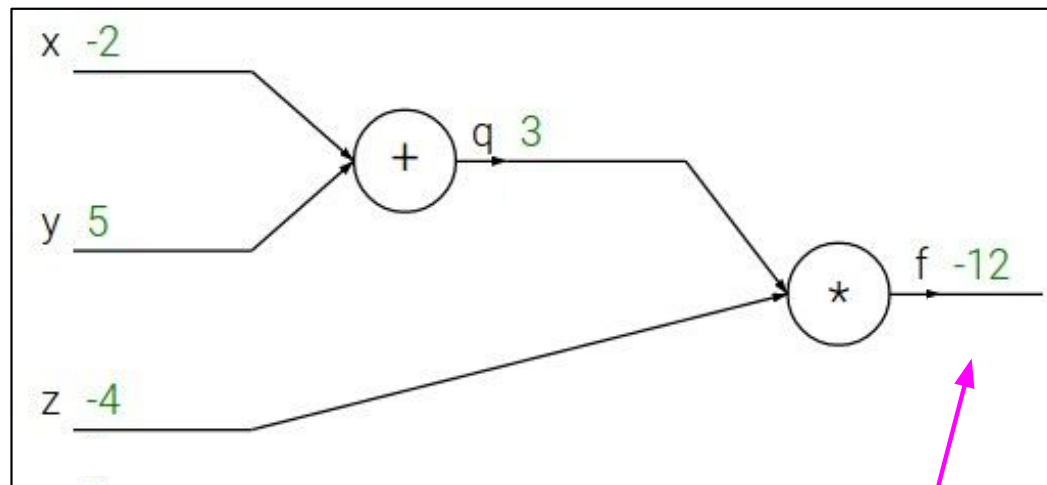
$$f(x, y, z) = (x + y)z$$

e.g. $x = -2$, $y = 5$, $z = -4$

$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \quad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



$$\frac{\partial f}{\partial f}$$

Backpropagation: a simple example

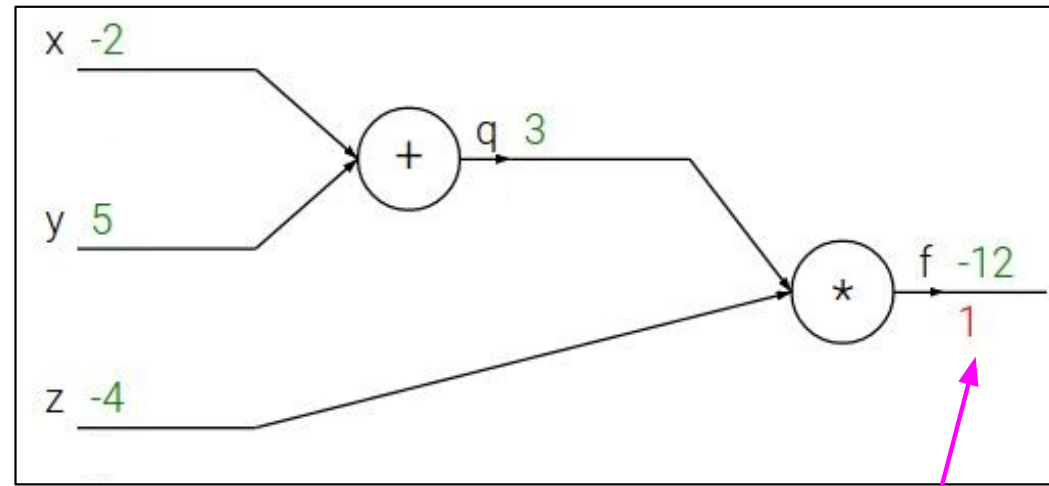
$$f(x, y, z) = (x + y)z$$

e.g. $x = -2$, $y = 5$, $z = -4$

$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \quad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



$$\frac{\partial f}{\partial f}$$

A pink arrow points from this box to the output f of the multiplication node in the computational graph above.

Backpropagation: a simple example

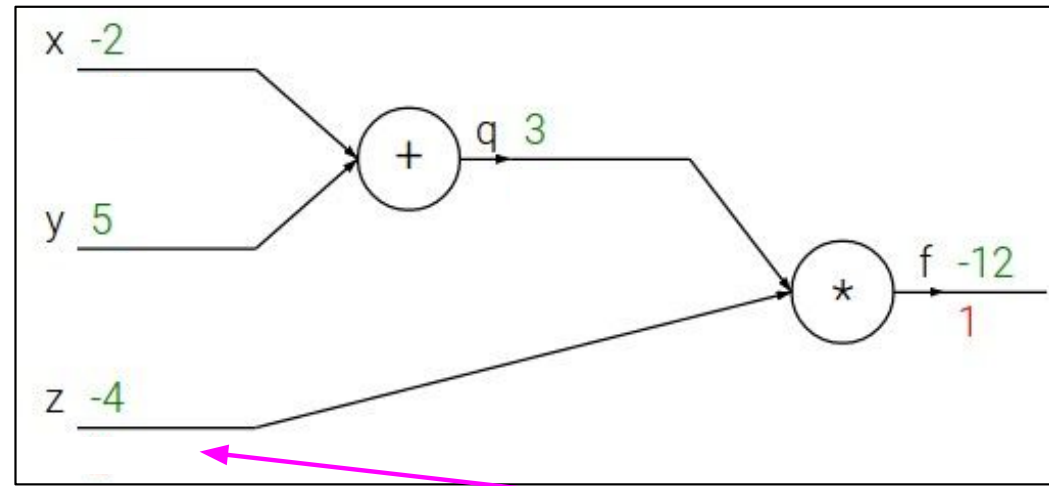
$$f(x, y, z) = (x + y)z$$

e.g. $x = -2$, $y = 5$, $z = -4$

$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \quad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



$$\frac{\partial f}{\partial z}$$

A magenta arrow points from this box to the z input line of the multiplication node in the computational graph above.

Backpropagation: a simple example

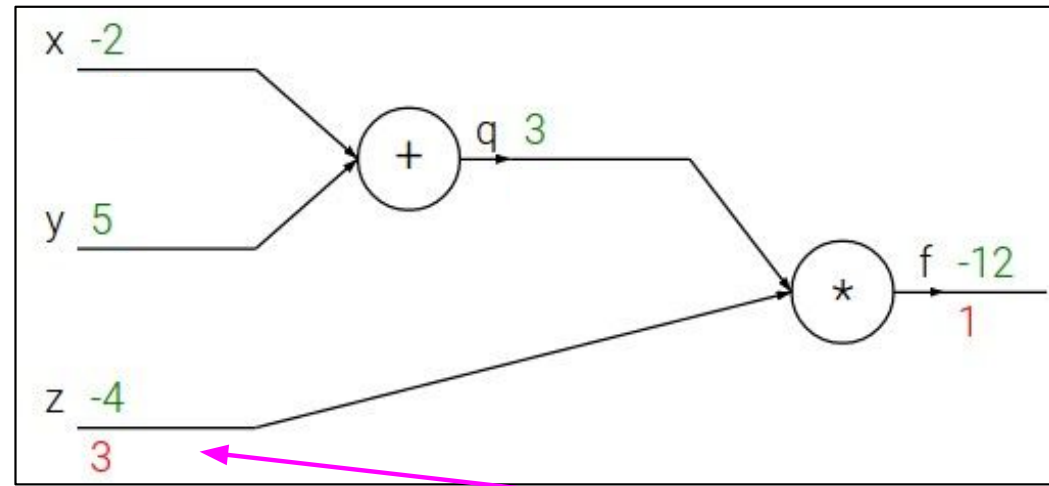
$$f(x, y, z) = (x + y)z$$

e.g. $x = -2$, $y = 5$, $z = -4$

$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \quad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



$$\frac{\partial f}{\partial z}$$

Backpropagation: a simple example

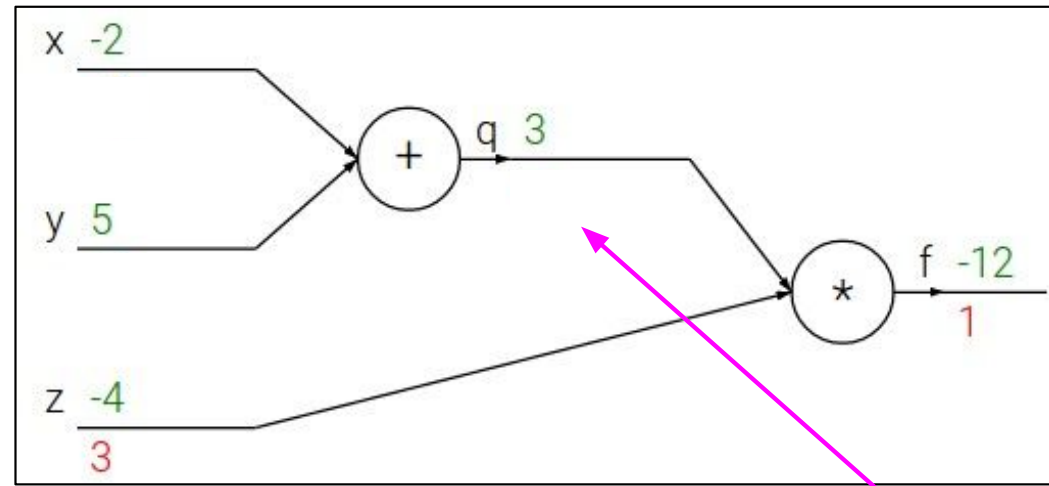
$$f(x, y, z) = (x + y)z$$

e.g. $x = -2$, $y = 5$, $z = -4$

$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \quad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



$$\frac{\partial f}{\partial q}$$

Backpropagation: a simple example

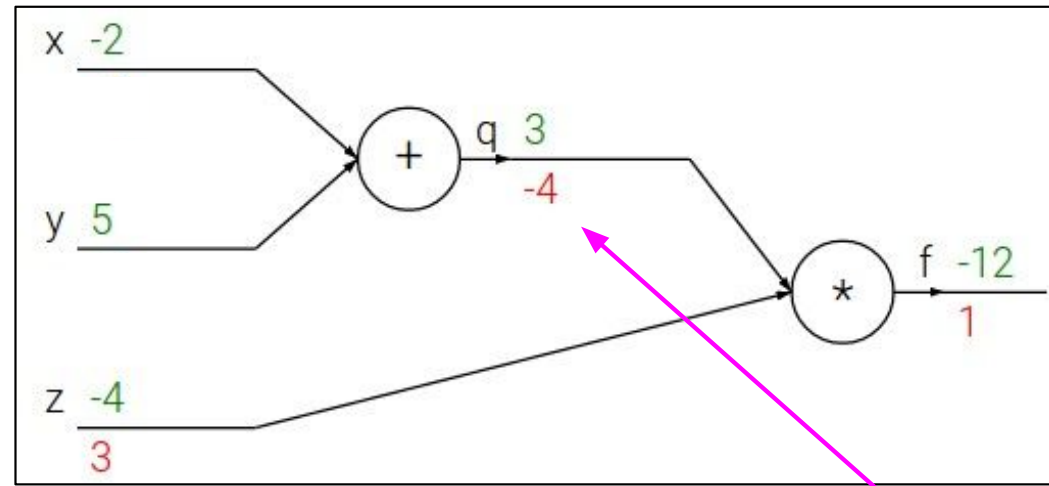
$$f(x, y, z) = (x + y)z$$

e.g. $x = -2$, $y = 5$, $z = -4$

$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \quad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



$$\frac{\partial f}{\partial q}$$

Backpropagation: a simple example

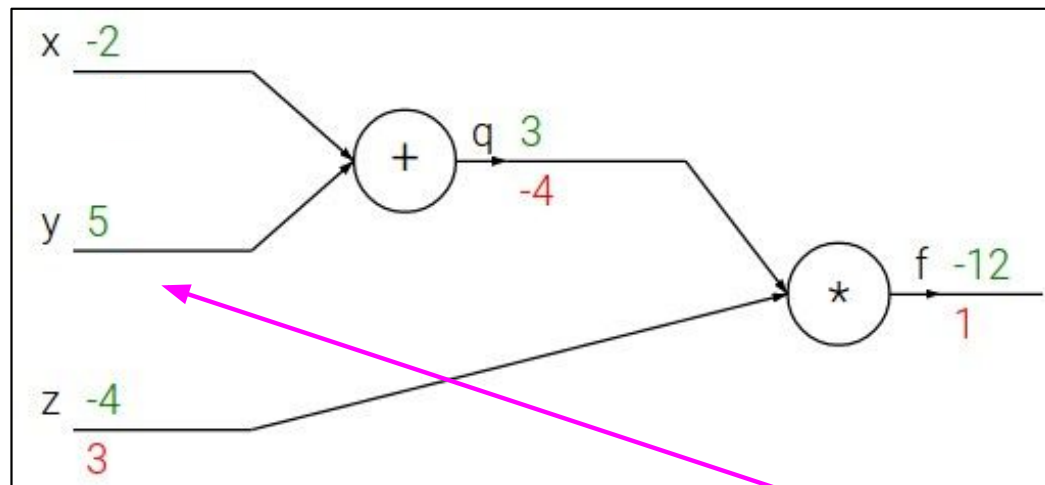
$$f(x, y, z) = (x + y)z$$

e.g. $x = -2, y = 5, z = -4$

$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \quad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



$$\frac{\partial f}{\partial y}$$

Chain rule:

$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial y}$$

Upstream
gradient

Local
gradient

Backpropagation: a simple example

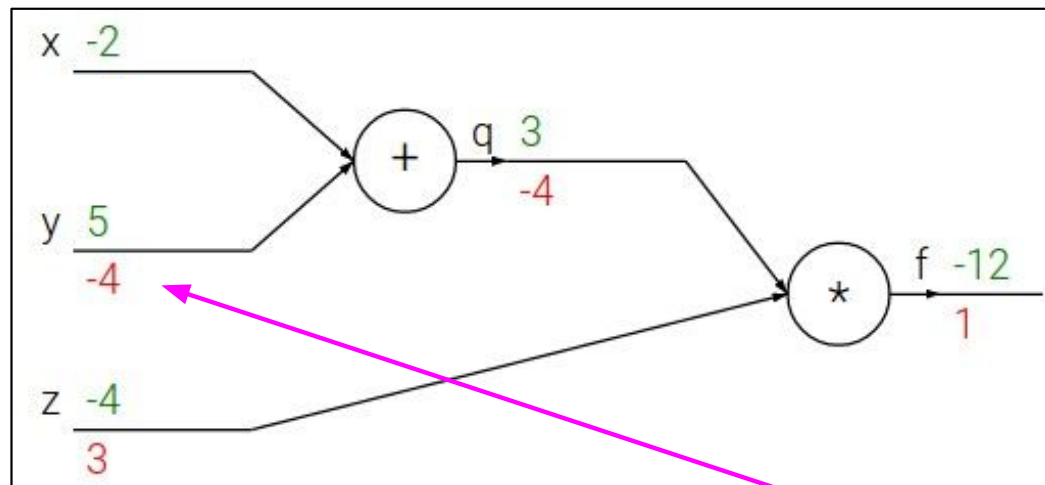
$$f(x, y, z) = (x + y)z$$

e.g. $x = -2$, $y = 5$, $z = -4$

$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \quad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



$$\frac{\partial f}{\partial y}$$

Chain rule:

$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial y}$$

Upstream
gradient

Local
gradient

Backpropagation: a simple example

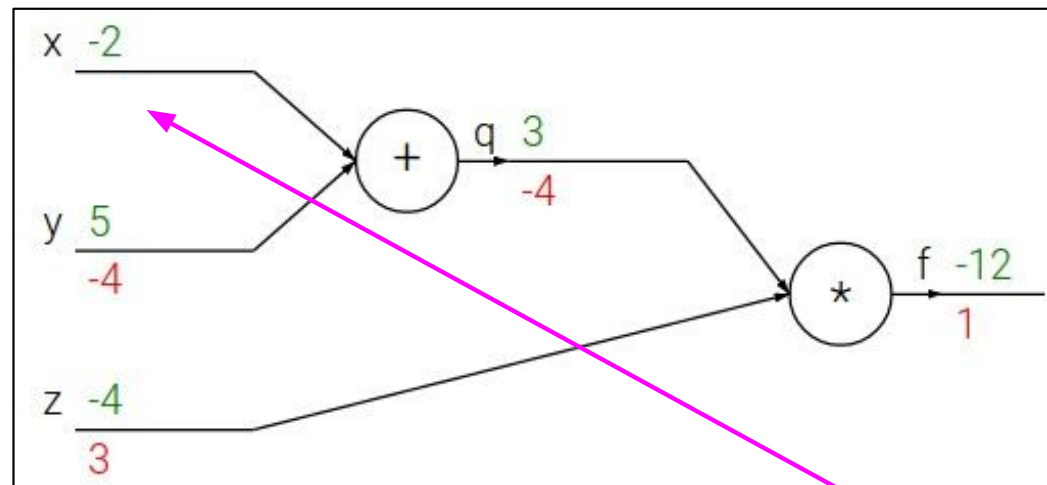
$$f(x, y, z) = (x + y)z$$

e.g. $x = -2$, $y = 5$, $z = -4$

$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \quad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



Chain rule:

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial x}$$

Upstream
gradient

Local
gradient

$$\frac{\partial f}{\partial x}$$

Backpropagation: a simple example

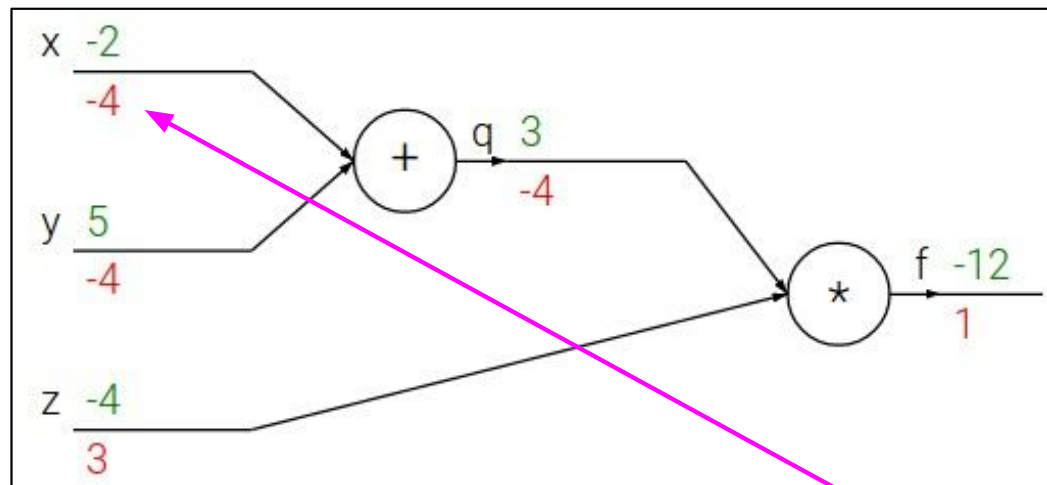
$$f(x, y, z) = (x + y)z$$

e.g. $x = -2$, $y = 5$, $z = -4$

$$q = x + y \quad \frac{\partial q}{\partial x} = 1, \frac{\partial q}{\partial y} = 1$$

$$f = qz \quad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

Want: $\frac{\partial f}{\partial x}$, $\frac{\partial f}{\partial y}$, $\frac{\partial f}{\partial z}$



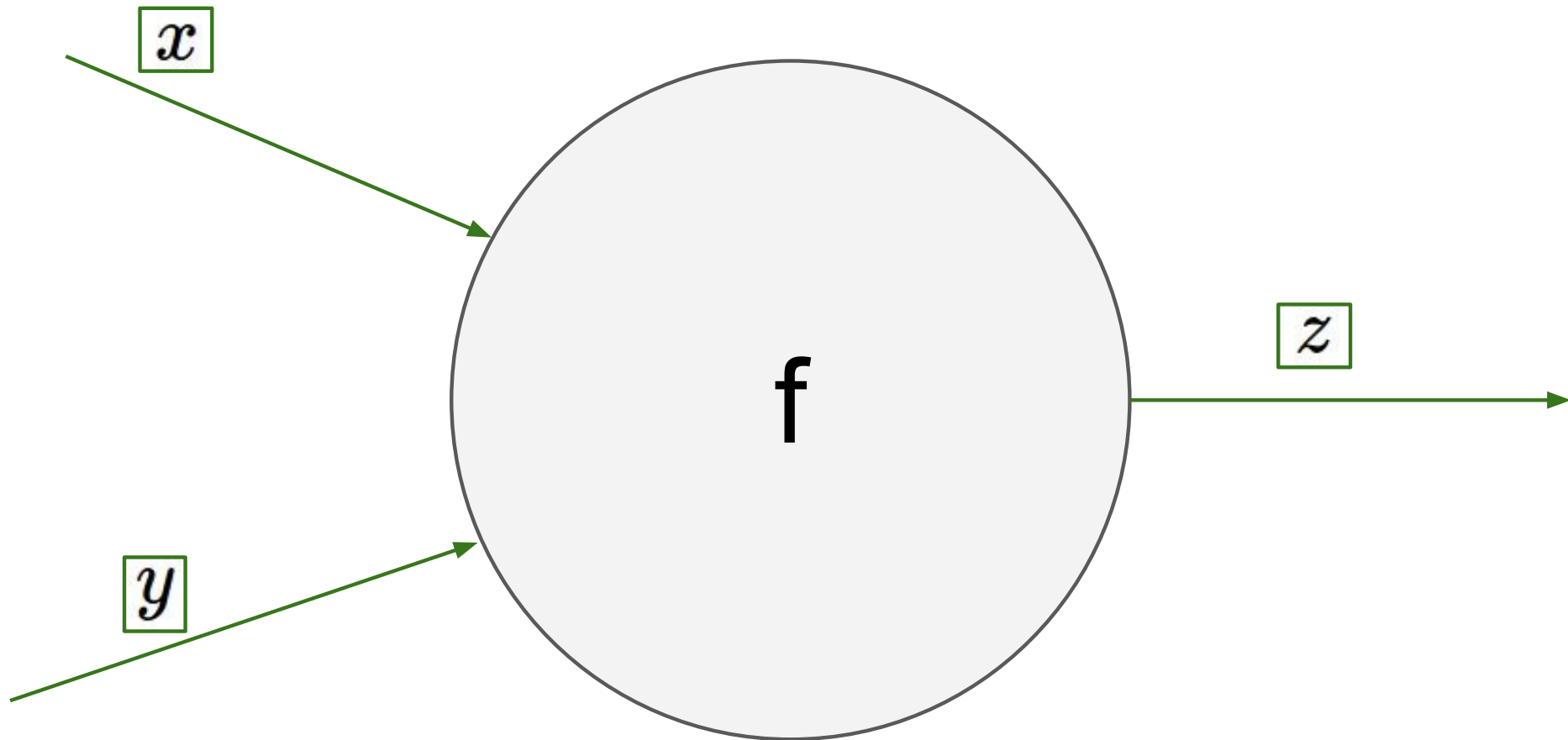
$$\frac{\partial f}{\partial x}$$

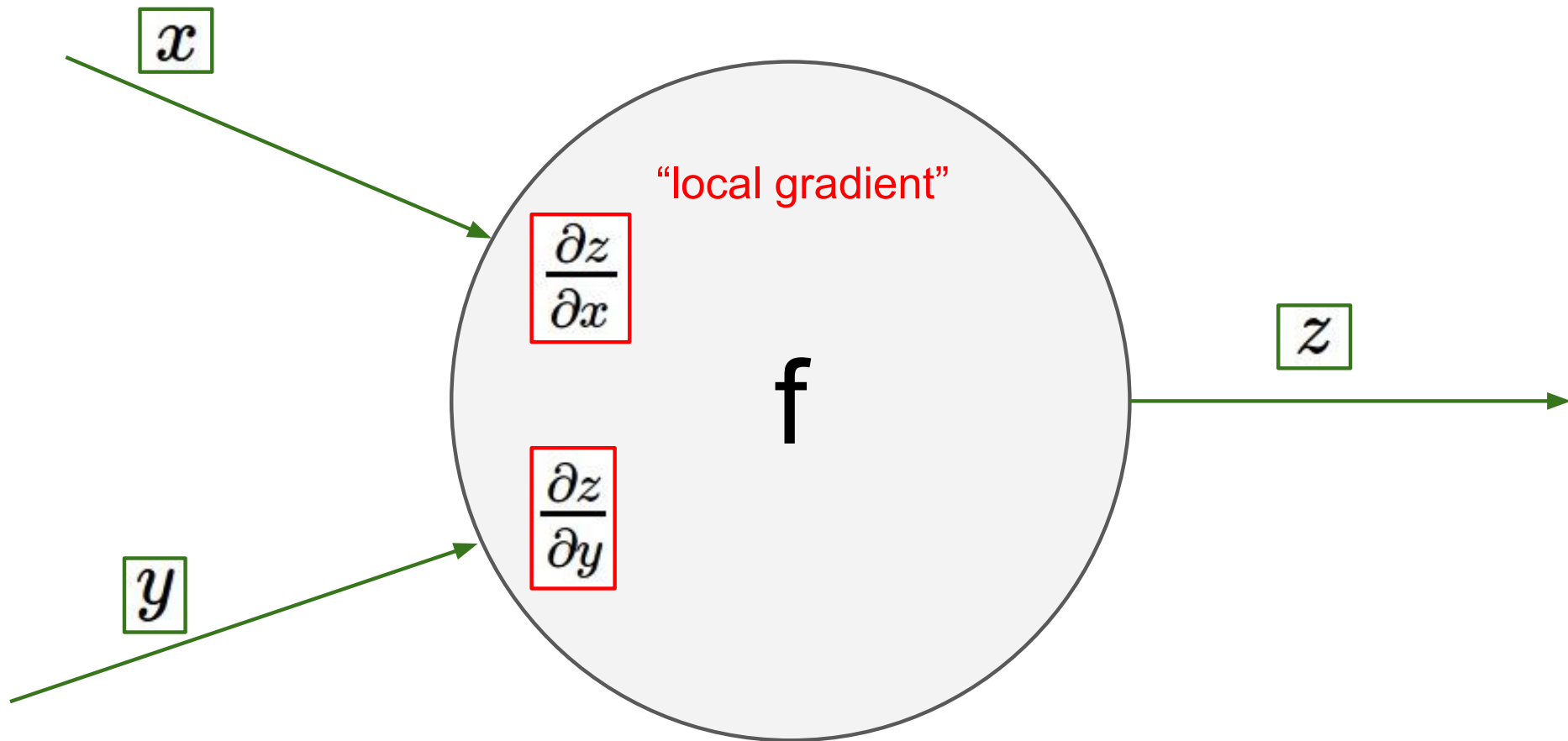
Chain rule:

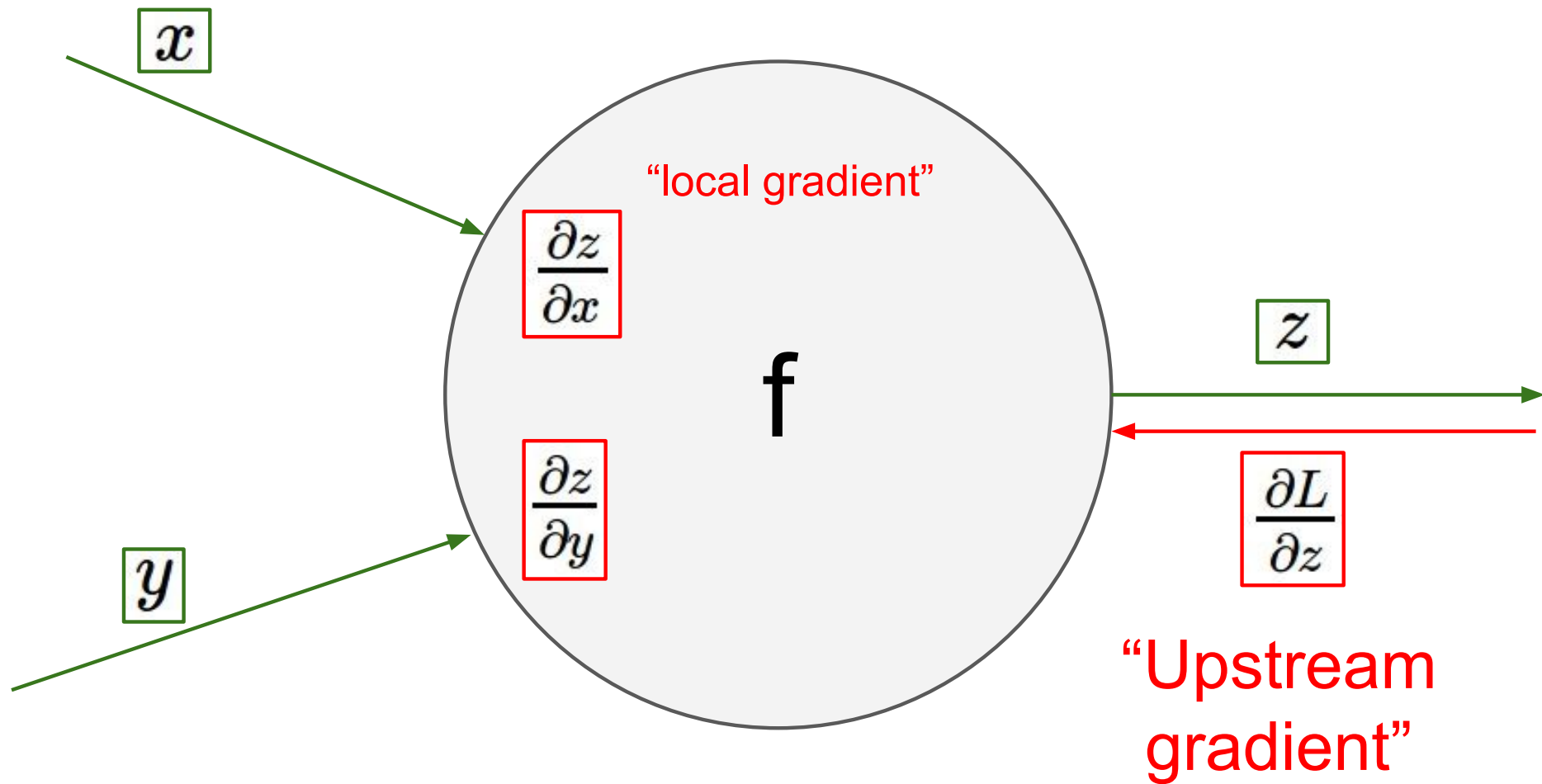
$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial x}$$

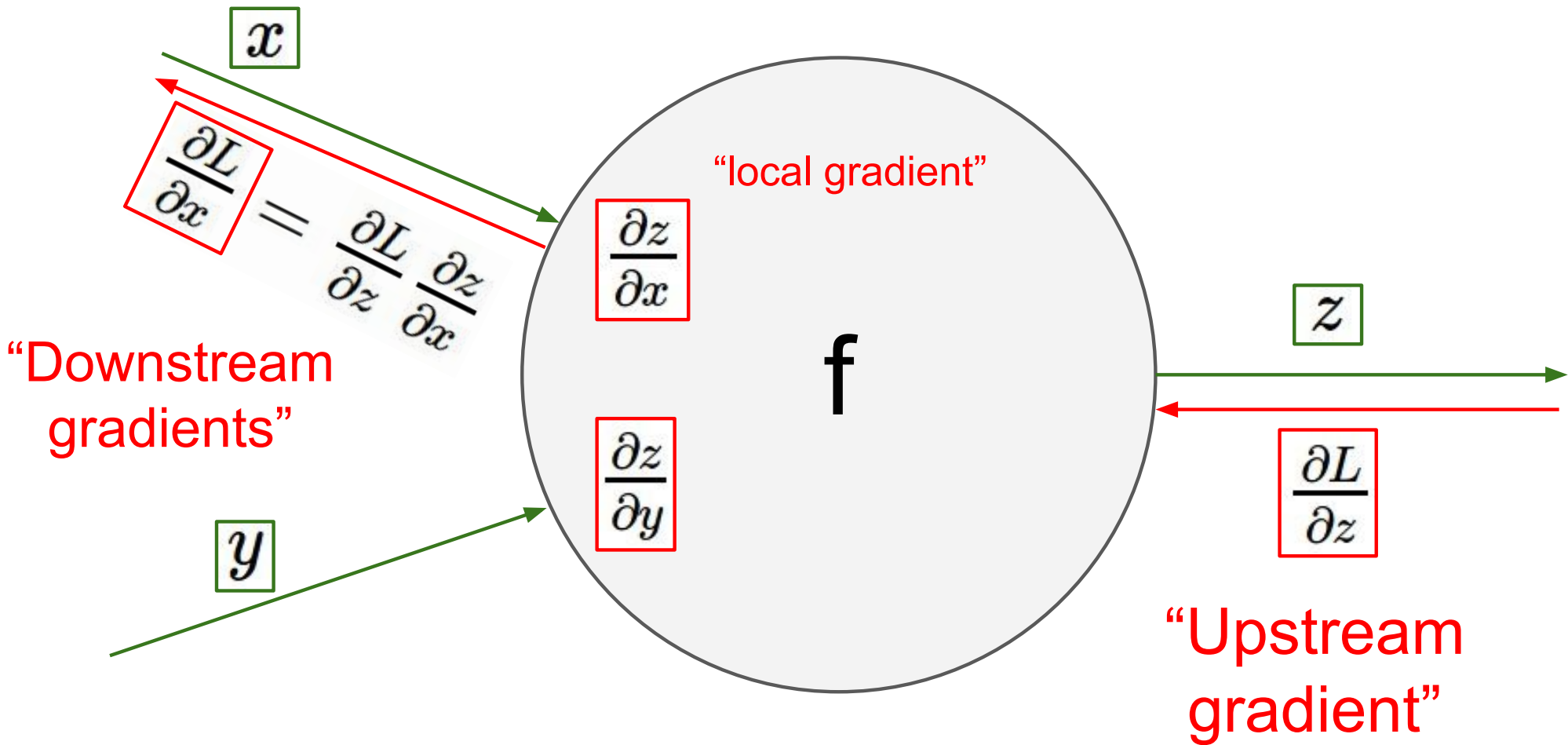
Upstream
gradient

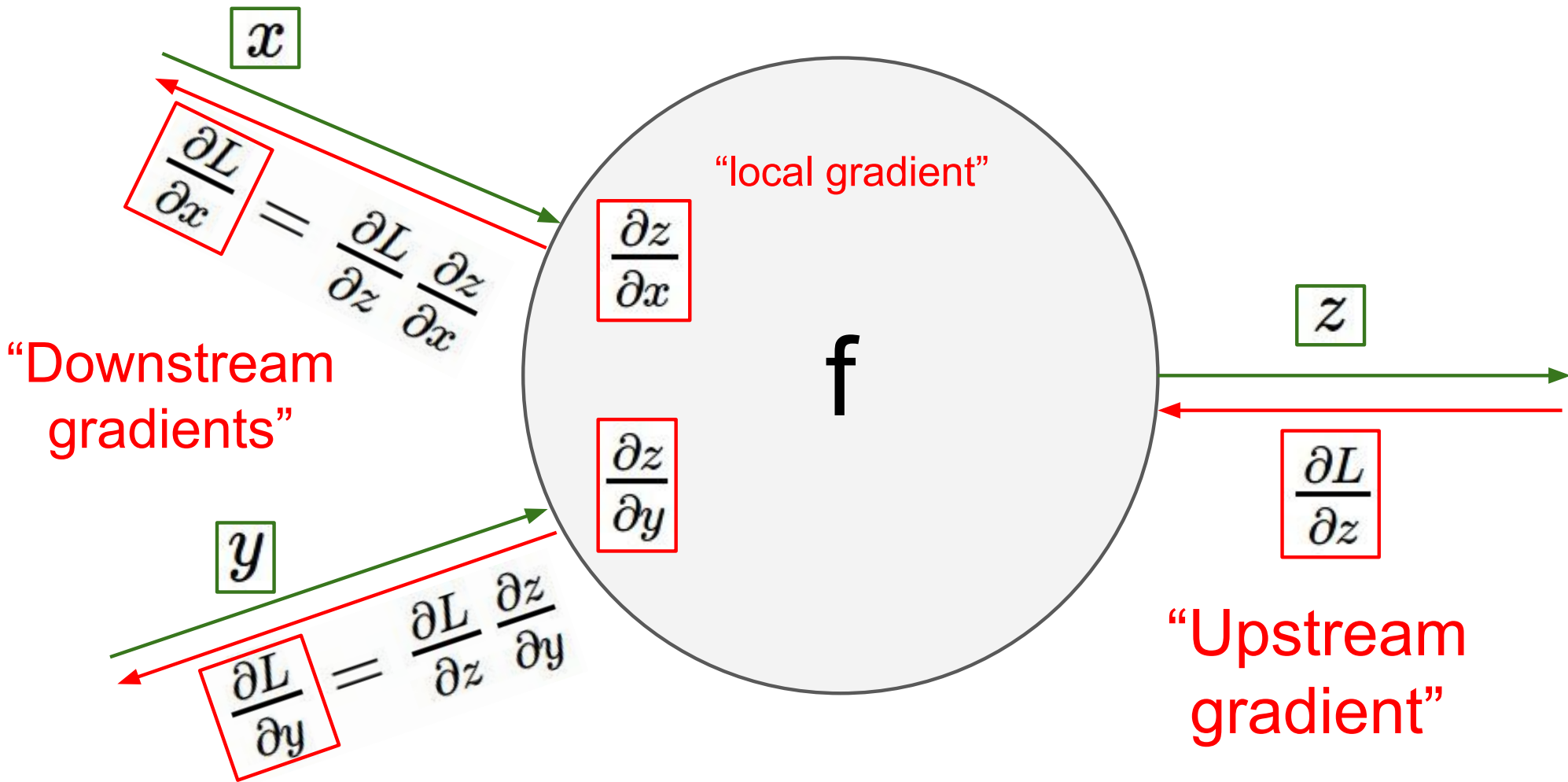
Local
gradient

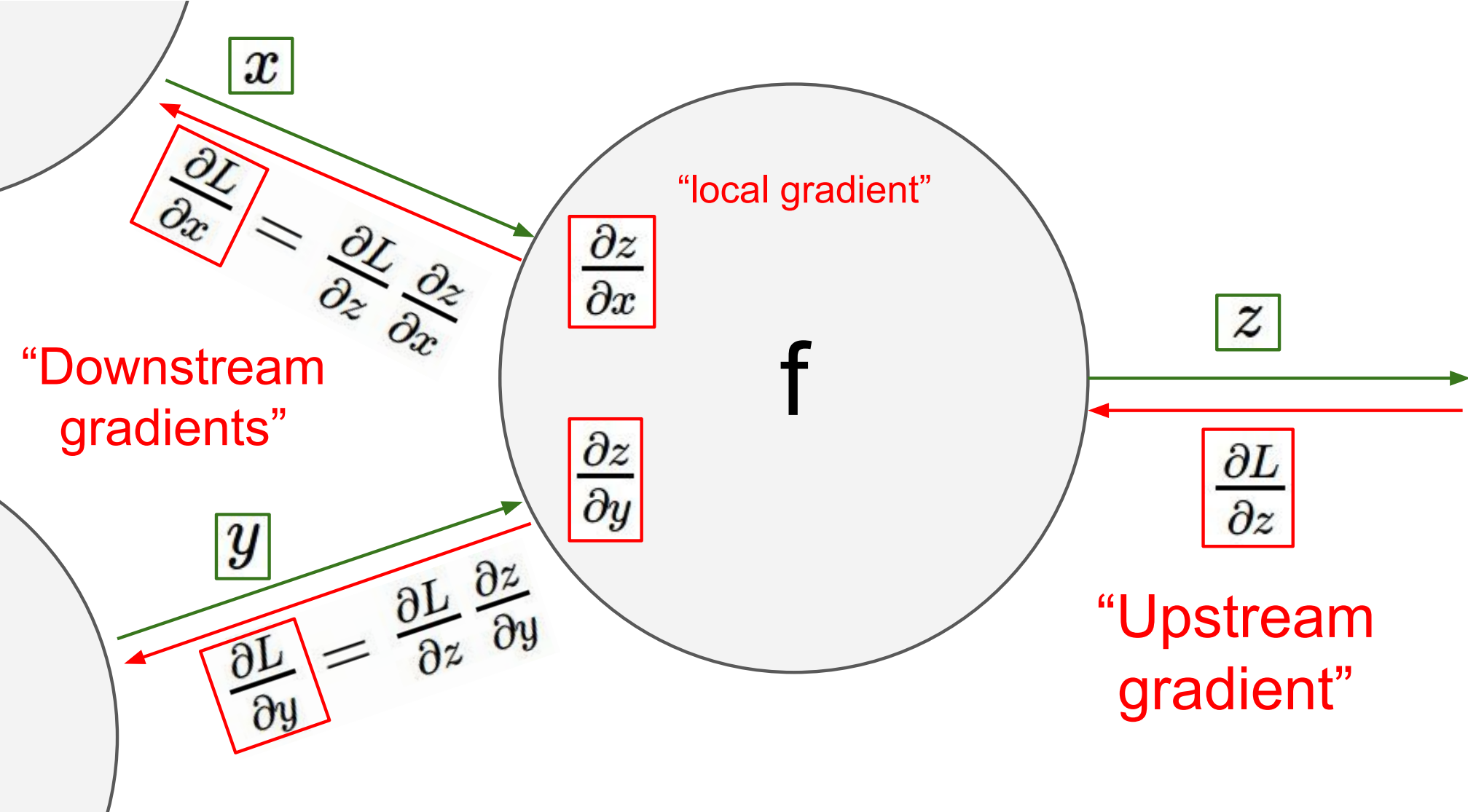






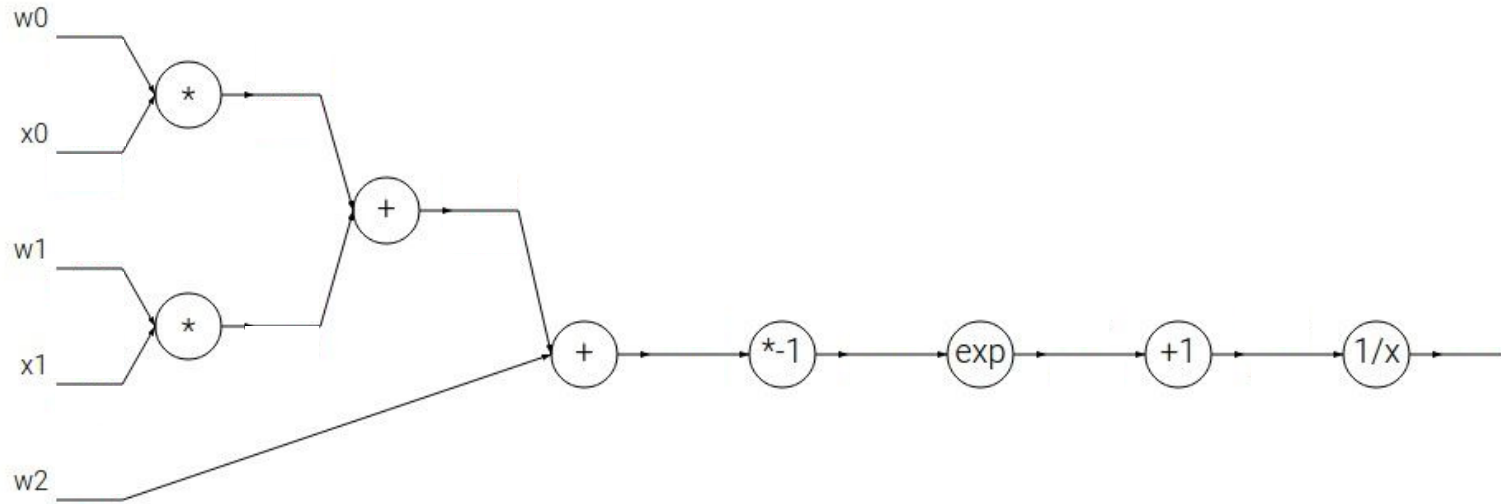






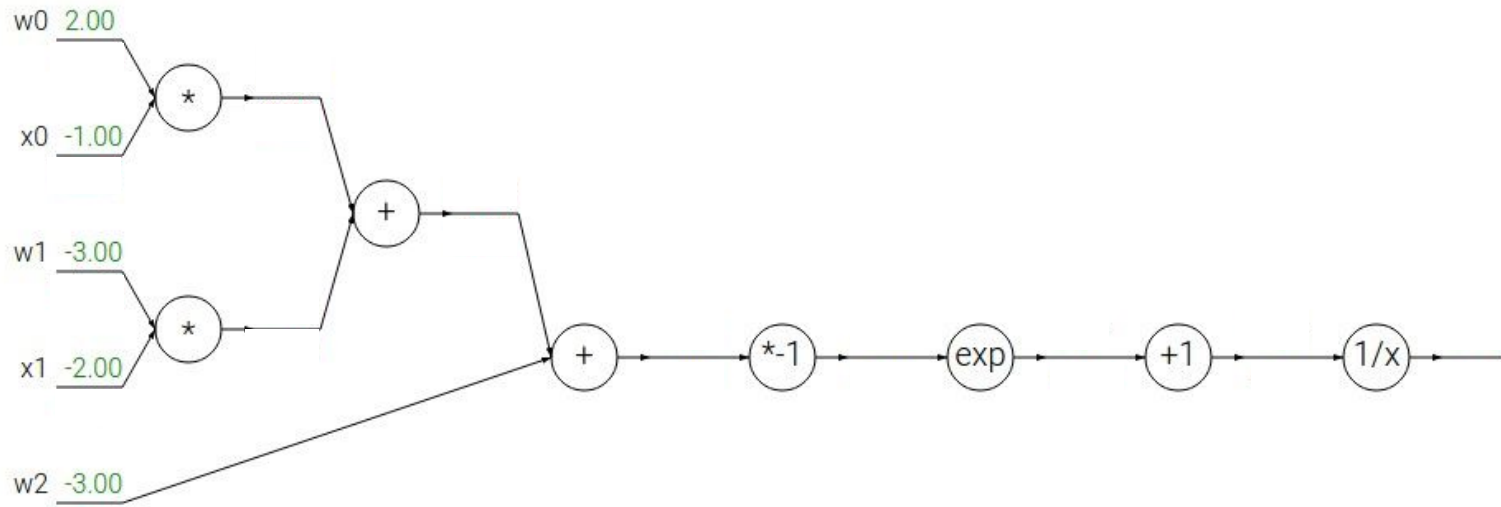
Another example:

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



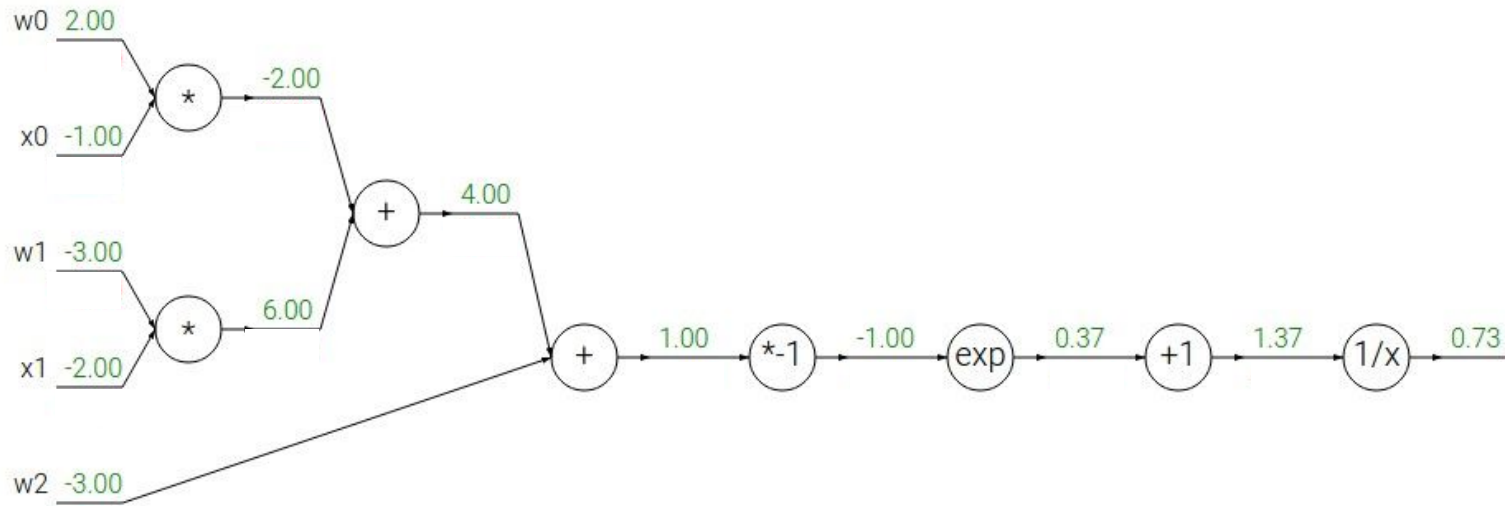
Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$



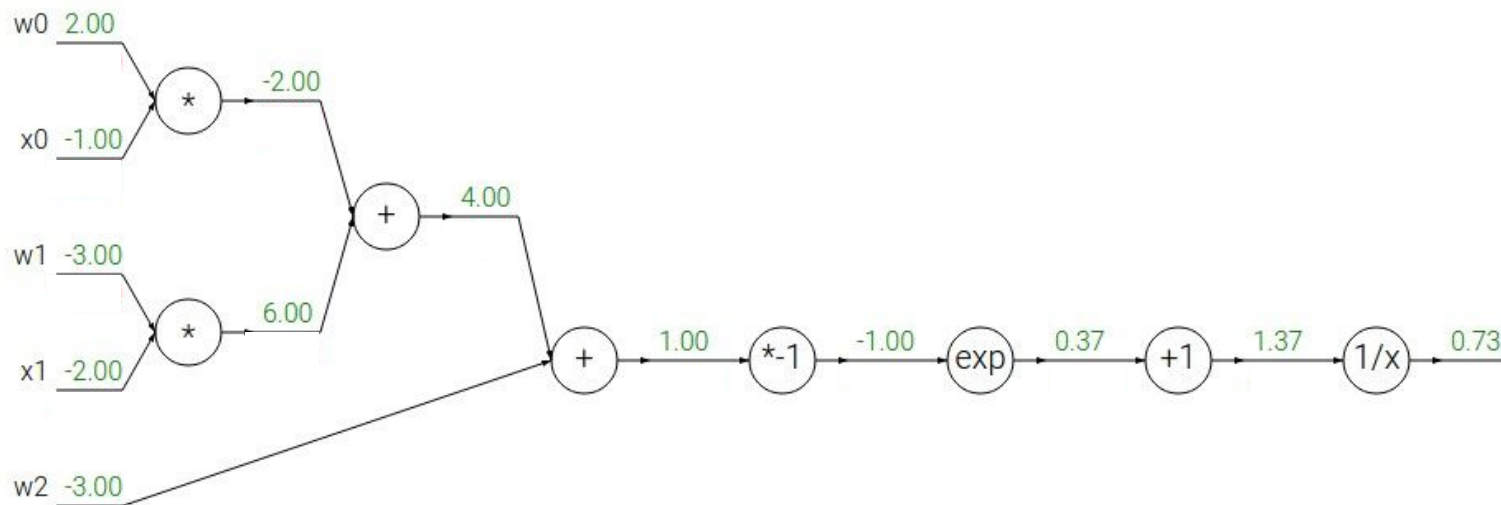
Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$



Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$



$$f(x) = e^x$$

→

$$\frac{df}{dx} = e^x$$

$$f_a(x) = ax$$

→

$$\frac{df}{dx} = a$$

$$f(x) = \frac{1}{x}$$

→

$$\frac{df}{dx} = -1/x^2$$

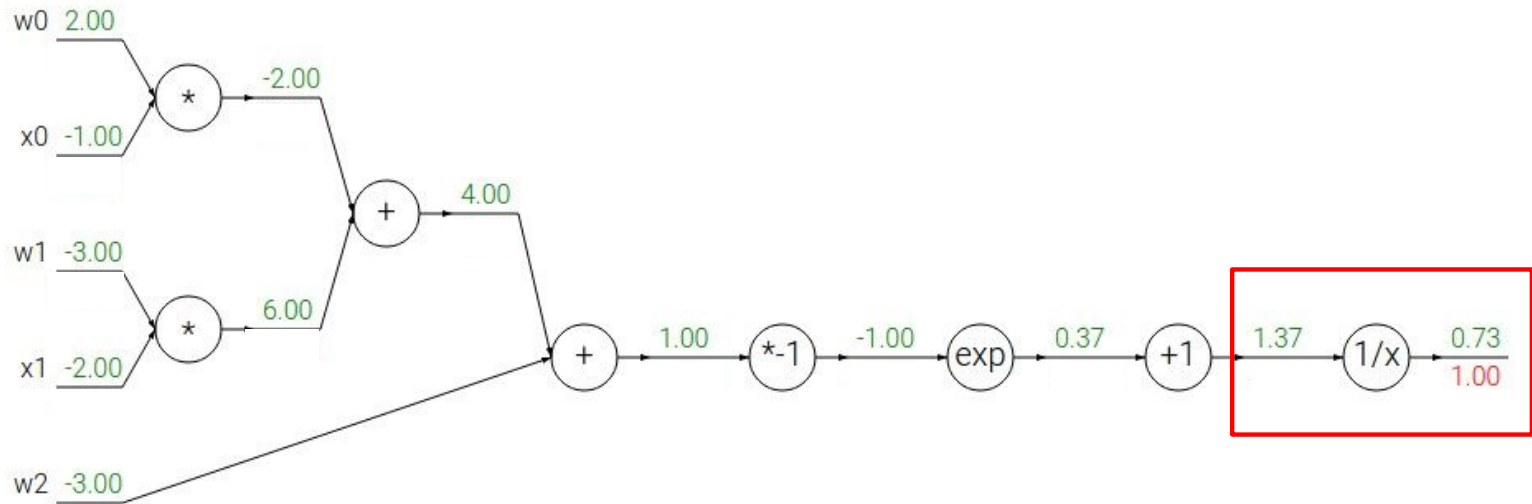
$$f_c(x) = c + x$$

→

$$\frac{df}{dx} = 1$$

Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$



$$f(x) = e^x$$

→

$$\frac{df}{dx} = e^x$$

$$f_a(x) = ax$$

→

$$\frac{df}{dx} = a$$

$$f(x) = \frac{1}{x}$$

→

$$\frac{df}{dx} = -1/x^2$$

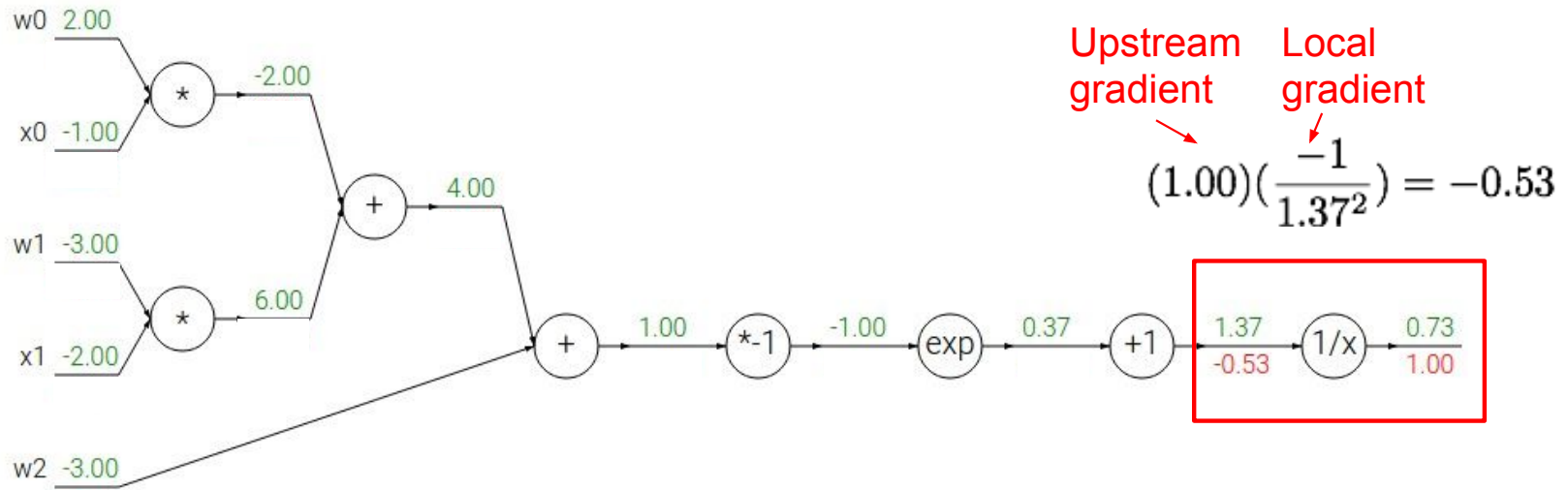
$$f_c(x) = c + x$$

→

$$\frac{df}{dx} = 1$$

Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$



$$f(x) = e^x$$

→

$$\frac{df}{dx} = e^x$$

$$f_a(x) = ax$$

→

$$\frac{df}{dx} = a$$

$$f(x) = \frac{1}{x}$$

→

$$\frac{df}{dx} = -1/x^2$$

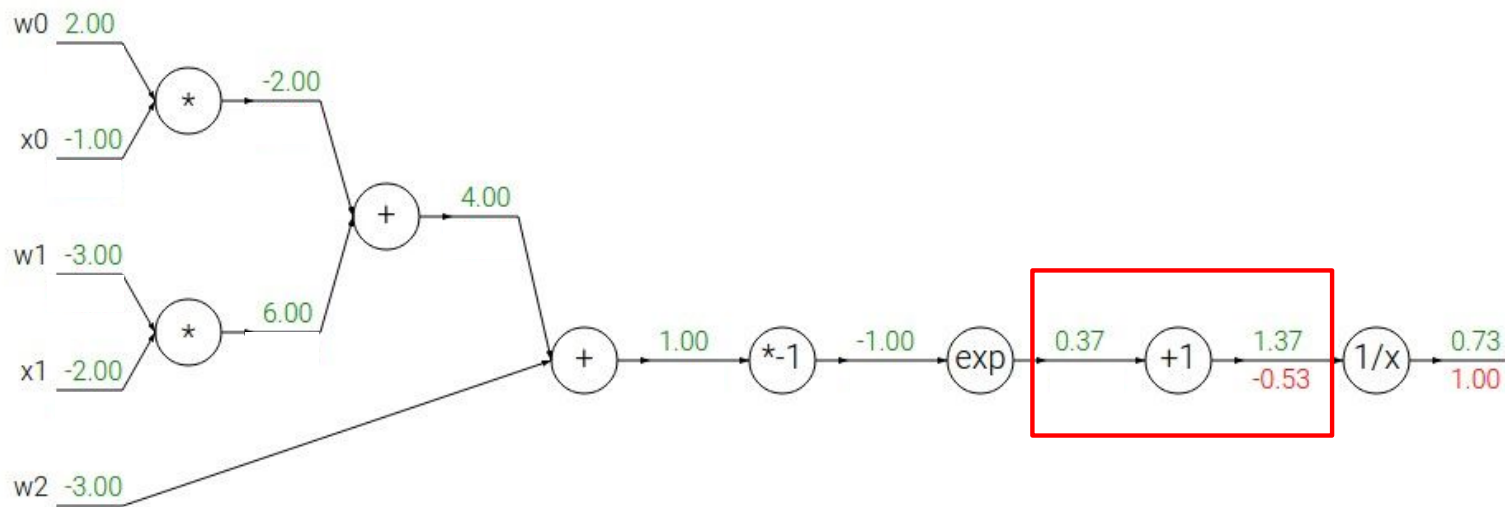
$$f_c(x) = c + x$$

→

$$\frac{df}{dx} = 1$$

Another example:

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



$$f(x) = e^x$$

\rightarrow

$$\frac{df}{dx} = e^x$$

$$f_a(x) = ax$$

\rightarrow

$$\frac{df}{dx} = a$$

$$f(x) = \frac{1}{x}$$

\rightarrow

$$\frac{df}{dx} = -1/x^2$$

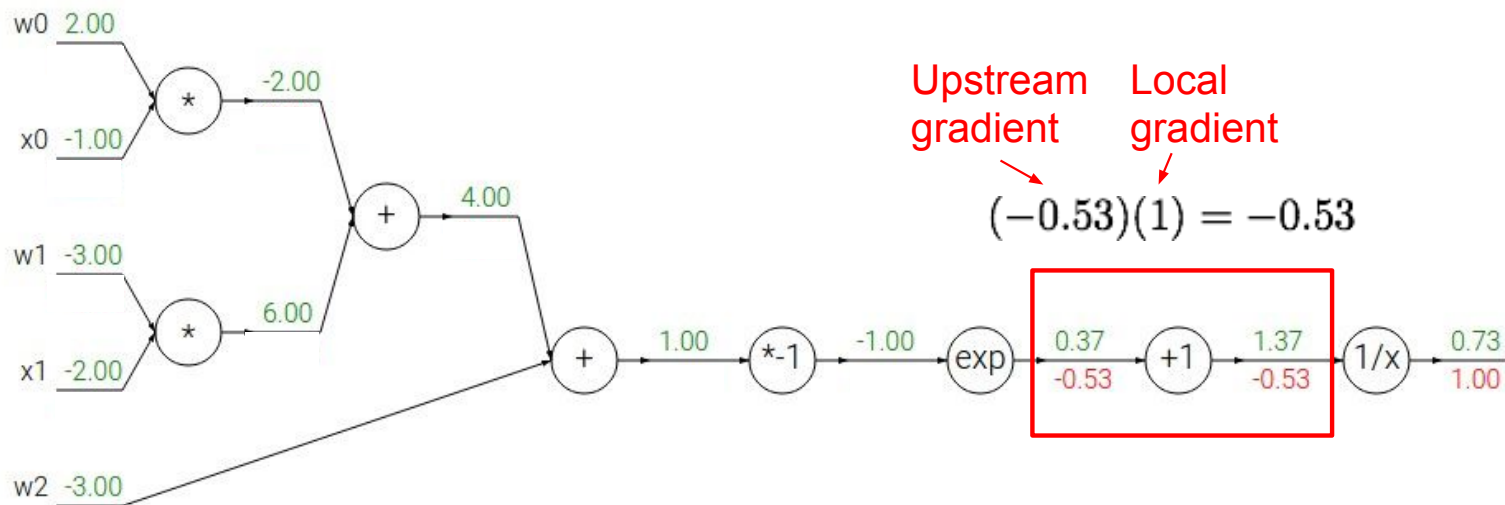
$$f_c(x) = c + x$$

\rightarrow

$$\frac{df}{dx} = 1$$

Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$



$$f(x) = e^x$$

→

$$\frac{df}{dx} = e^x$$

$$f_a(x) = ax$$

→

$$\frac{df}{dx} = a$$

$$f(x) = \frac{1}{x}$$

→

$$\frac{df}{dx} = -1/x^2$$

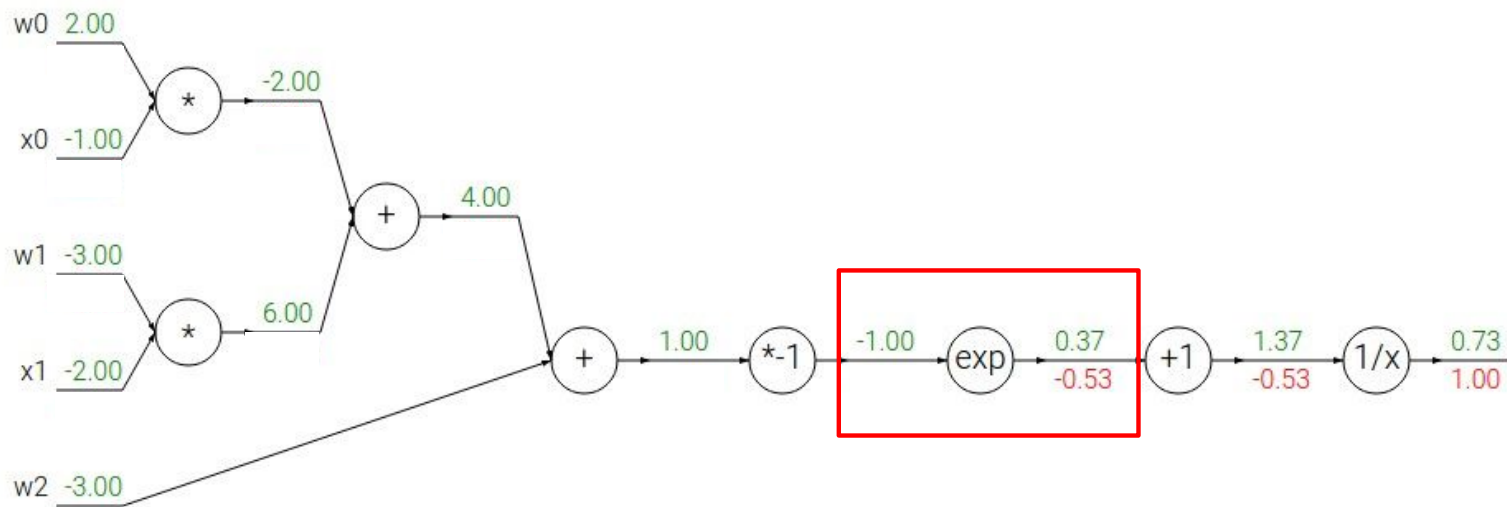
$$f_c(x) = c + x$$

→

$$\frac{df}{dx} = 1$$

Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$



$$f(x) = e^x$$

→

$$\frac{df}{dx} = e^x$$

$$f_a(x) = ax$$

→

$$\frac{df}{dx} = a$$

$$f(x) = \frac{1}{x}$$

→

$$\frac{df}{dx} = -1/x^2$$

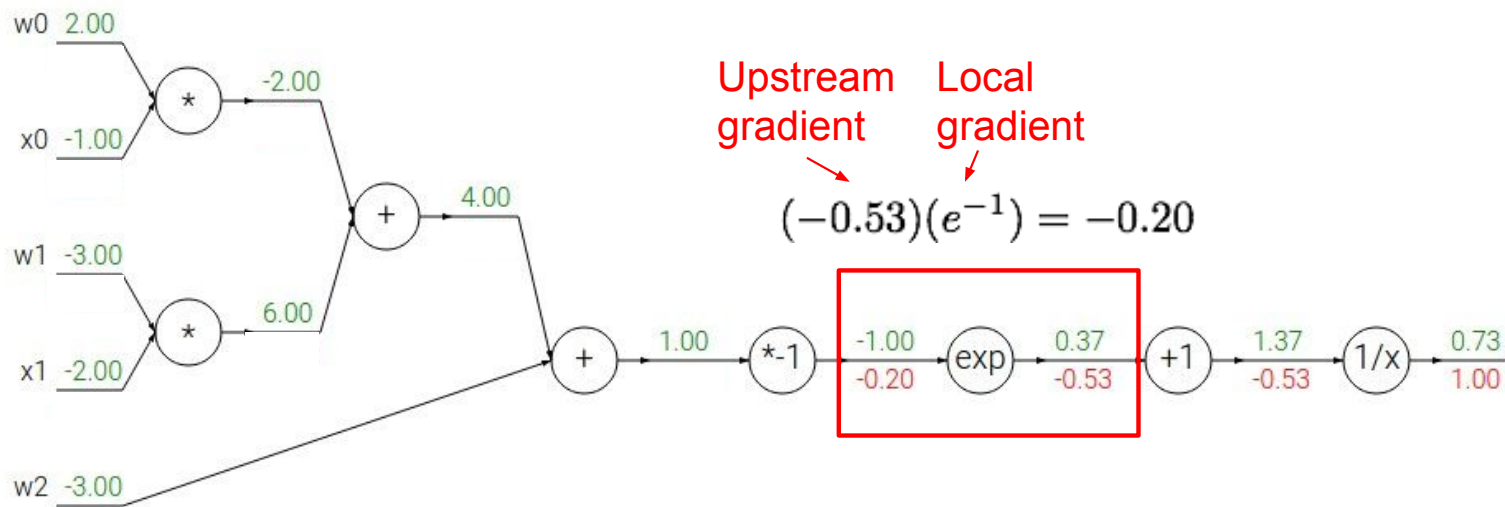
$$f_c(x) = c + x$$

→

$$\frac{df}{dx} = 1$$

Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$



$$\boxed{f(x) = e^x \rightarrow \frac{df}{dx} = e^x}$$

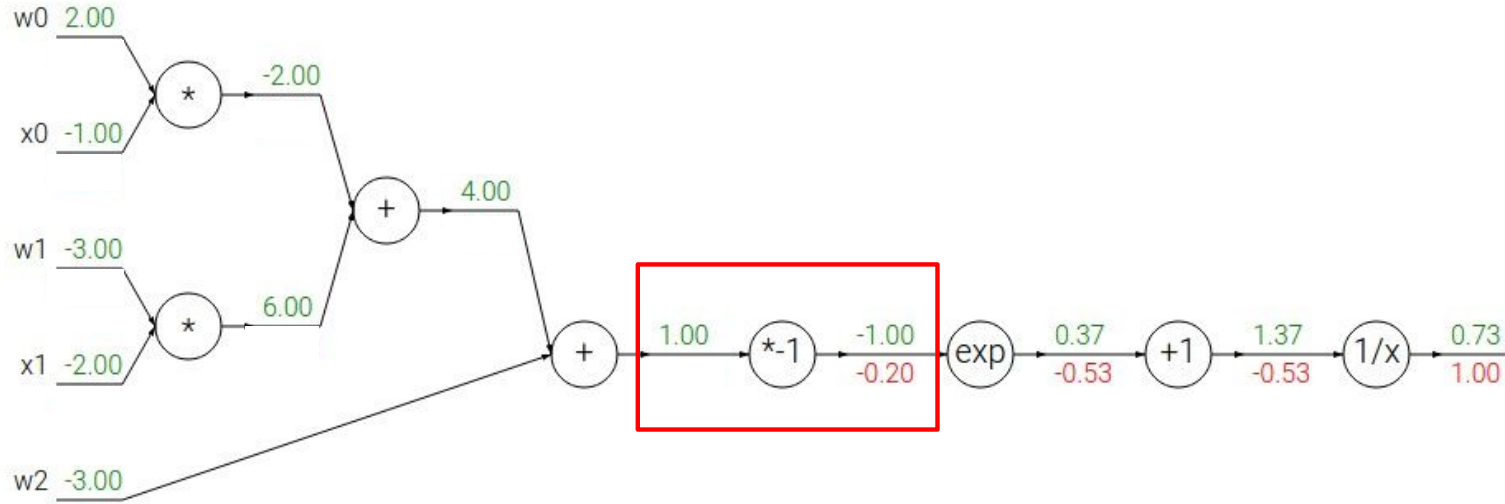
$$f_a(x) = ax \rightarrow \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \rightarrow \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \rightarrow \frac{df}{dx} = 1$$

Another example:

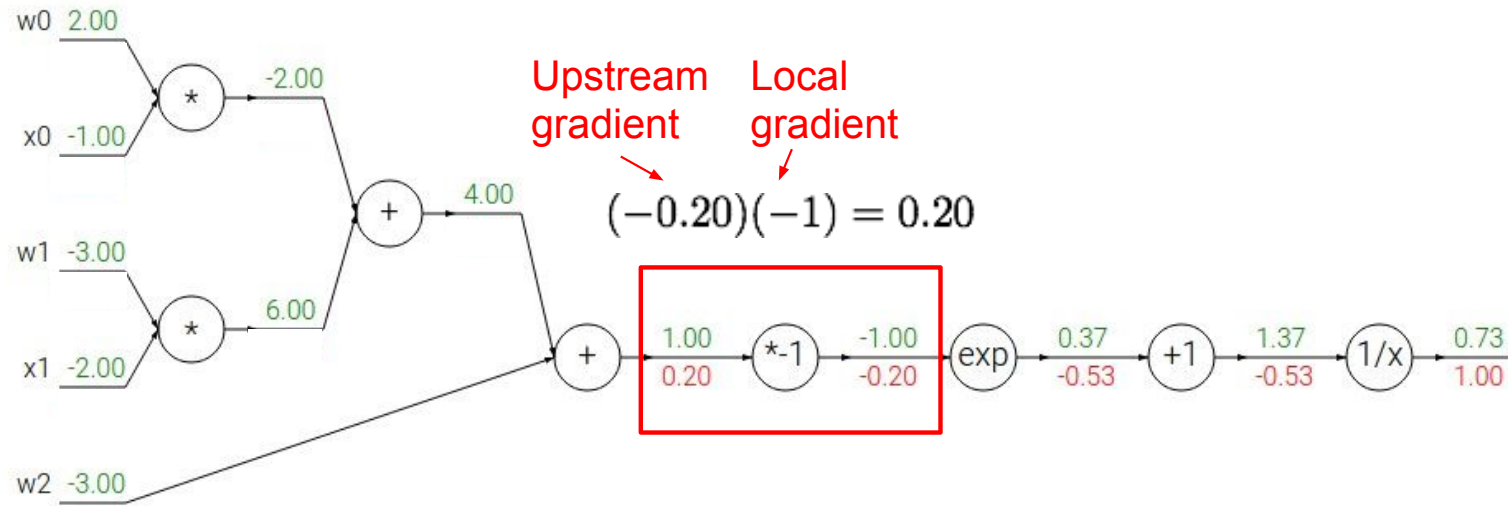
$$f(w, x) = \frac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2)}}$$



$f(x) = e^x$	\rightarrow	$\frac{df}{dx} = e^x$		$f(x) = \frac{1}{x}$	\rightarrow	$\frac{df}{dx} = -1/x^2$
$f_a(x) = ax$	\rightarrow	$\frac{df}{dx} = a$		$f_c(x) = c + x$	\rightarrow	$\frac{df}{dx} = 1$

Another example:

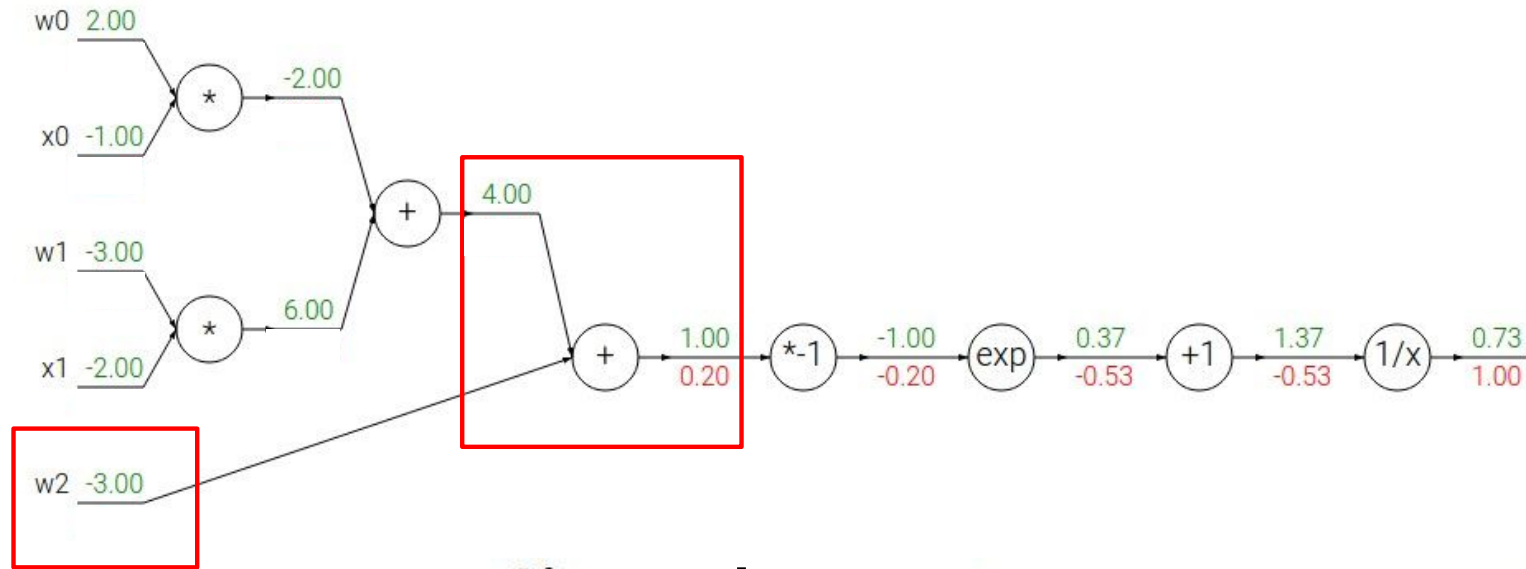
$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$



$f(x) = e^x$	\rightarrow	$\frac{df}{dx} = e^x$		$f(x) = \frac{1}{x}$	\rightarrow	$\frac{df}{dx} = -1/x^2$
$f_a(x) = ax$	\rightarrow	$\frac{df}{dx} = a$		$f_c(x) = c + x$	\rightarrow	$\frac{df}{dx} = 1$

Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$



$$f(x) = e^x$$

→

$$\frac{df}{dx} = e^x$$

$$f_a(x) = ax$$

→

$$\frac{df}{dx} = a$$

$$f(x) = \frac{1}{x}$$

→

$$\frac{df}{dx} = -1/x^2$$

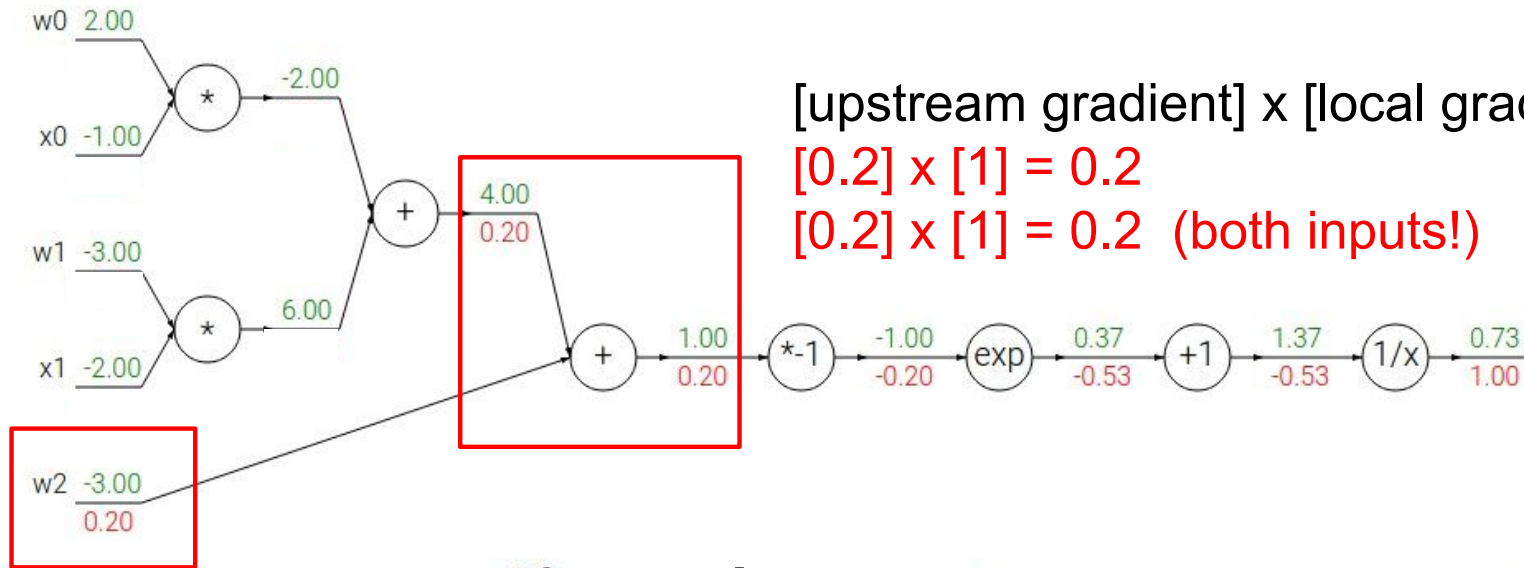
$$f_c(x) = c + x$$

→

$$\frac{df}{dx} = 1$$

Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$



[upstream gradient] x [local gradient]

$$[0.2] \times [1] = 0.2$$

$$[0.2] \times [1] = 0.2 \text{ (both inputs!)}$$

$$f(x) = e^x$$

→

$$\frac{df}{dx} = e^x$$

$$f_a(x) = ax$$

→

$$\frac{df}{dx} = a$$

$$f(x) = \frac{1}{x}$$

→

$$\frac{df}{dx} = -1/x^2$$

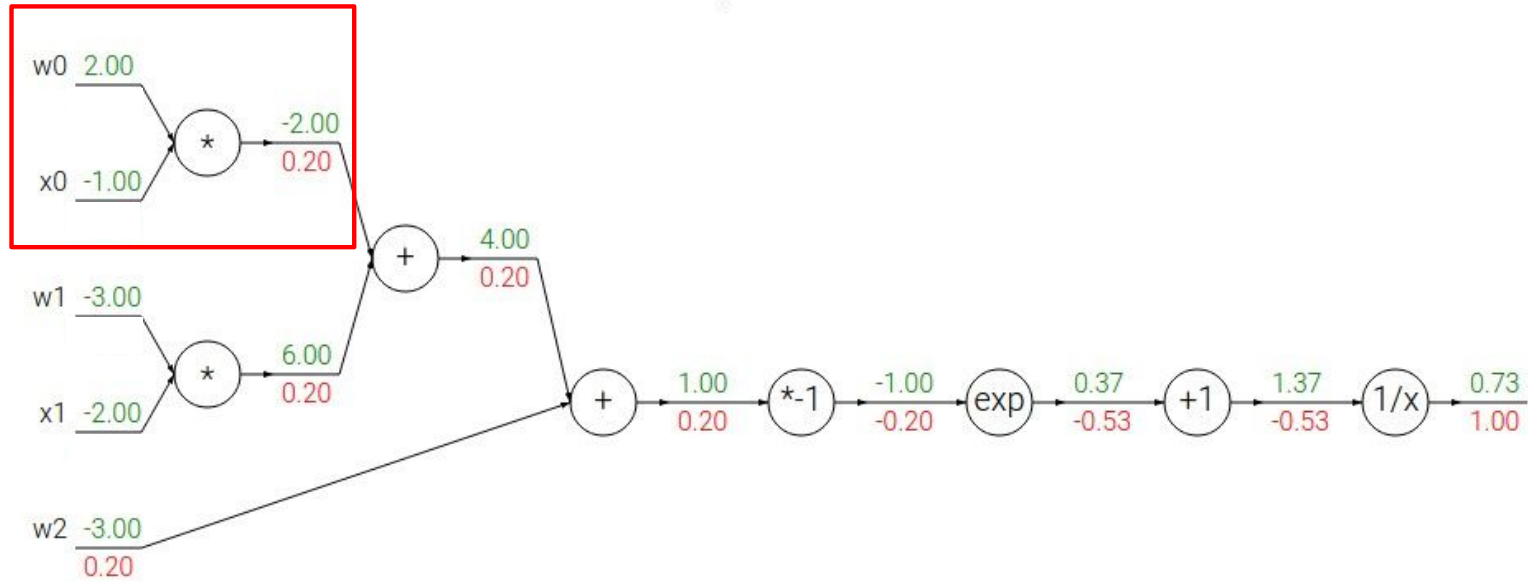
$$f_c(x) = c + x$$

→

$$\frac{df}{dx} = 1$$

Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$



$$f(x) = e^x$$

→

$$\frac{df}{dx} = e^x$$

$$f(x) = \frac{1}{x}$$

→

$$\frac{df}{dx} = -1/x^2$$

$$f_a(x) = ax$$

→

$$\frac{df}{dx} = a$$

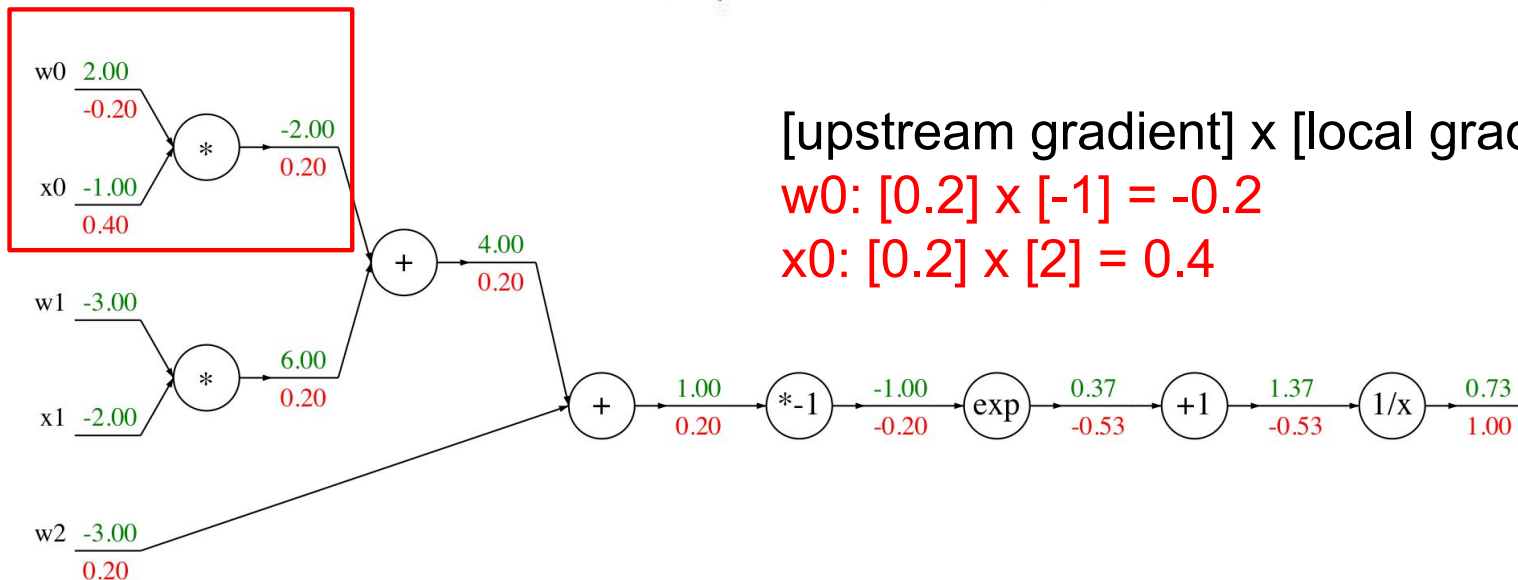
$$f_c(x) = c + x$$

→

$$\frac{df}{dx} = 1$$

Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2x_2)}}$$



$$f(x) = e^x$$

→

$$\frac{df}{dx} = e^x$$

$$f_a(x) = ax$$

→

$$\frac{df}{dx} = a$$

$$f(x) = \frac{1}{x}$$

→

$$\frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x$$

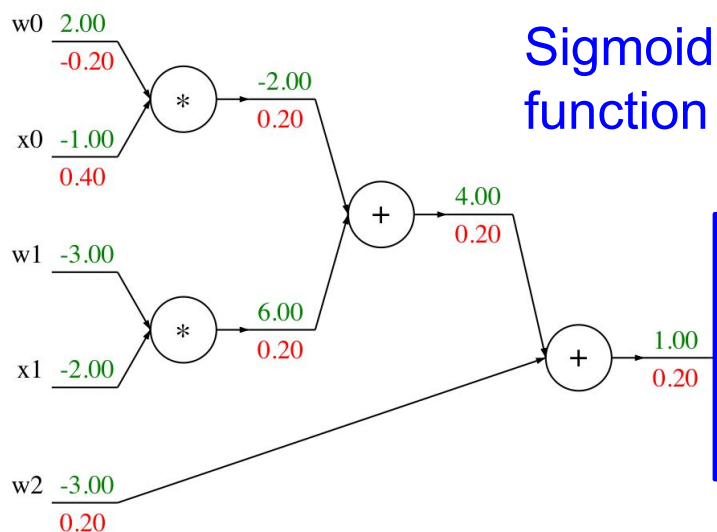
→

$$\frac{df}{dx} = 1$$

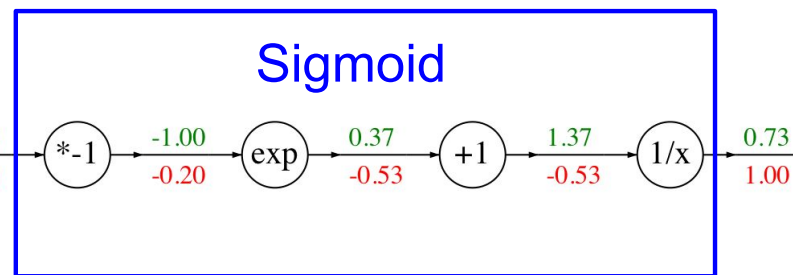
Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$

Computational graph representation may not be unique. Choose one where local gradients at each node can be easily expressed!



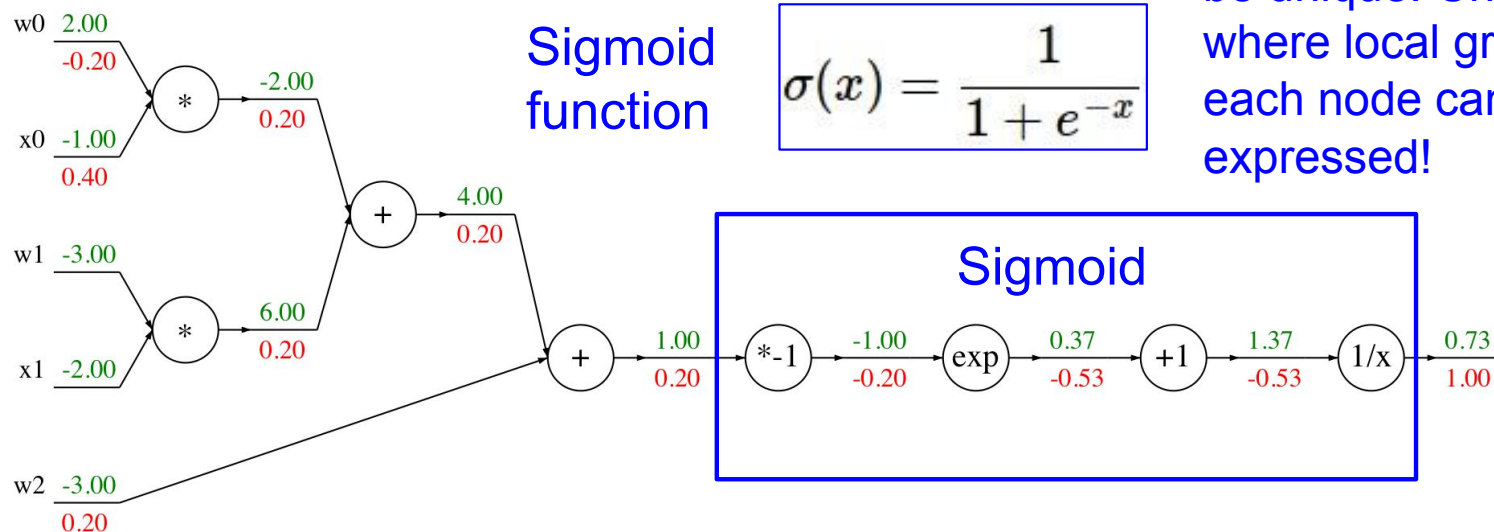
$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



Another example:

$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$

Computational graph representation may not be unique. Choose one where local gradients at each node can be easily expressed!



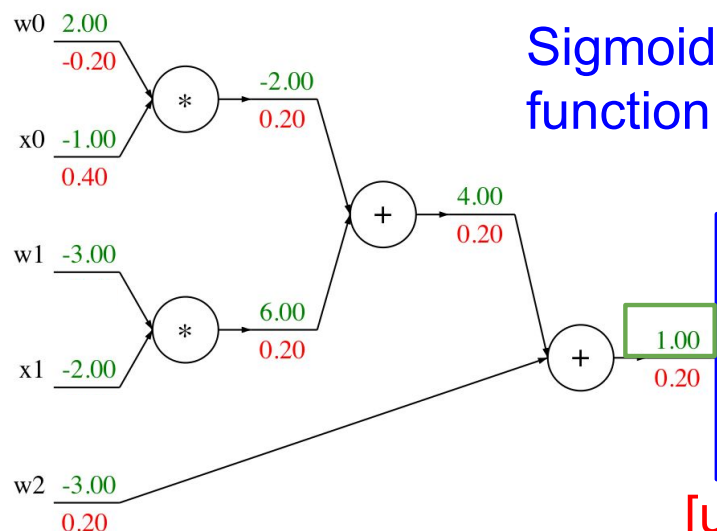
Sigmoid local gradient:

$$\frac{d\sigma(x)}{dx} = \frac{e^{-x}}{(1 + e^{-x})^2} = \left(\frac{1 + e^{-x} - 1}{1 + e^{-x}} \right) \left(\frac{1}{1 + e^{-x}} \right) = (1 - \sigma(x)) \sigma(x)$$

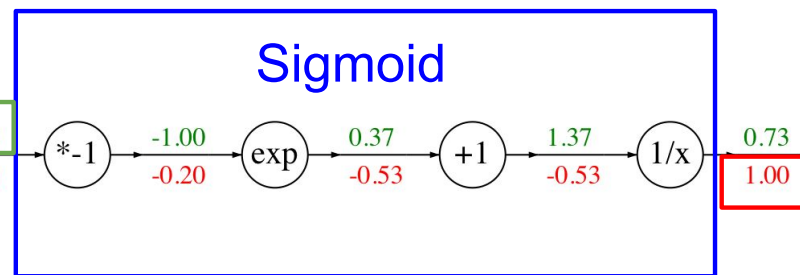
Another example:

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

Computational graph representation may not be unique. Choose one where local gradients at each node can be easily expressed!



$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



[upstream gradient] x [local gradient]
 $[1.00] \times [(1 - 1/(1+e^1)) (1/(1+e^1))] = 0.2$

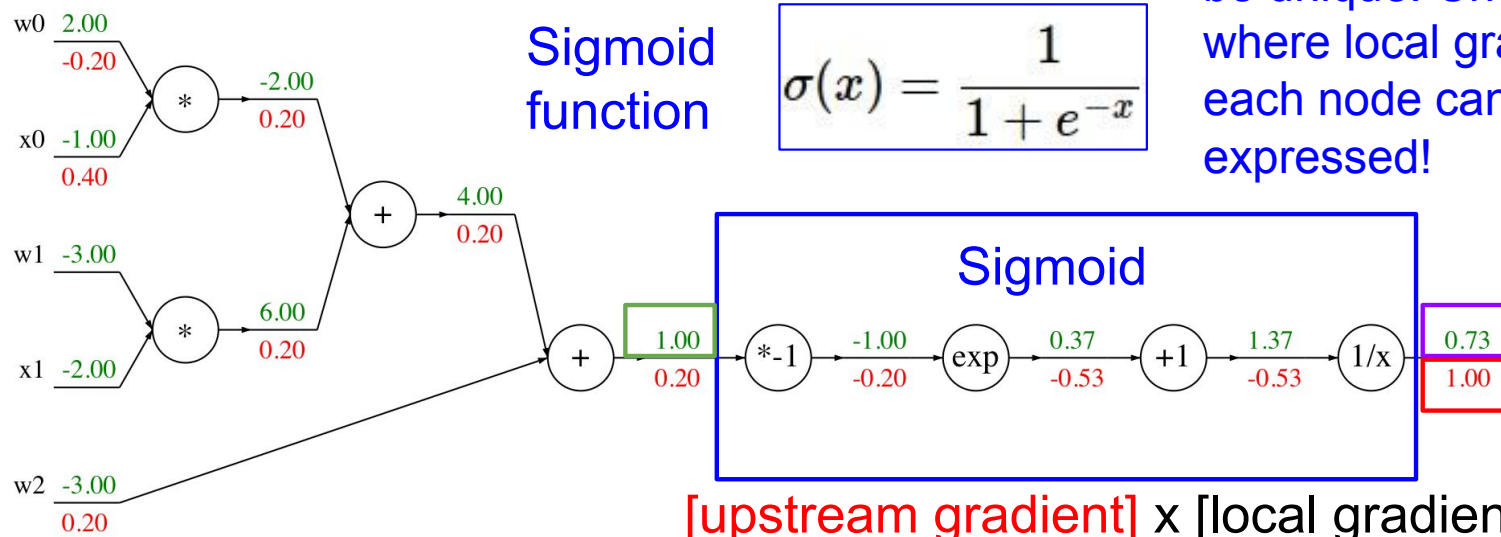
Sigmoid local gradient:

$$\frac{d\sigma(x)}{dx} = \frac{e^{-x}}{(1 + e^{-x})^2} = \left(\frac{1 + e^{-x} - 1}{1 + e^{-x}} \right) \left(\frac{1}{1 + e^{-x}} \right) = (1 - \sigma(x)) \sigma(x)$$

Another example:

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

Computational graph representation may not be unique. Choose one where local gradients at each node can be easily expressed!



Sigmoid function

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

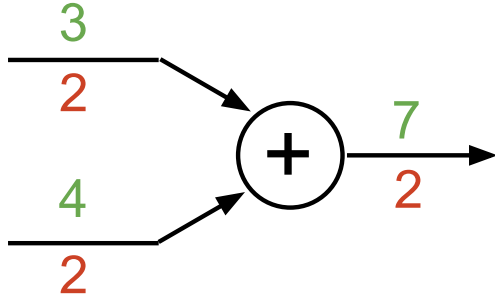
[upstream gradient] x [local gradient]
 $[1.00] \times [(1 - 0.73) (0.73)] = 0.2$

Sigmoid local gradient:

$$\frac{d\sigma(x)}{dx} = \frac{e^{-x}}{(1 + e^{-x})^2} = \left(\frac{1 + e^{-x} - 1}{1 + e^{-x}} \right) \left(\frac{1}{1 + e^{-x}} \right) = (1 - \sigma(x)) \sigma(x)$$

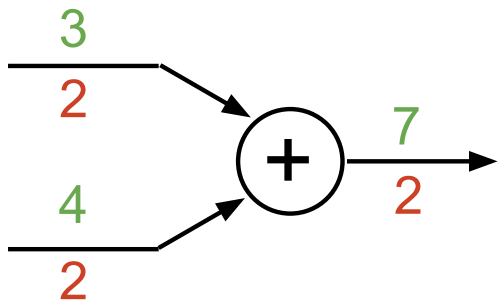
Patterns in gradient flow

add gate: gradient distributor

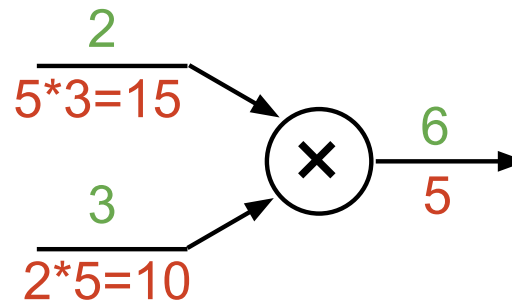


Patterns in gradient flow

add gate: gradient distributor

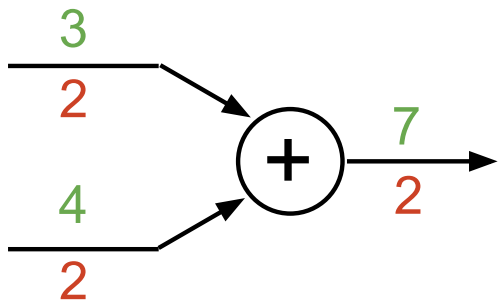


mul gate: “swap multiplier”

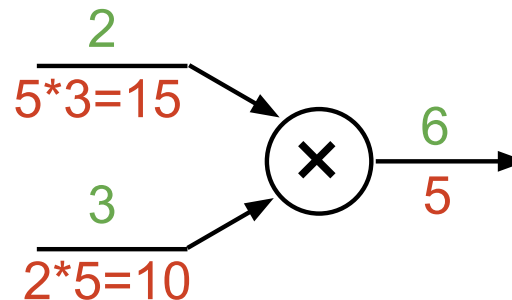


Patterns in gradient flow

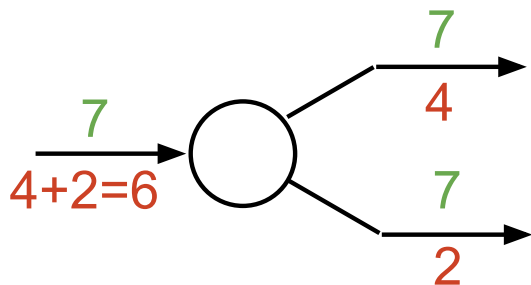
add gate: gradient distributor



mul gate: “swap multiplier”

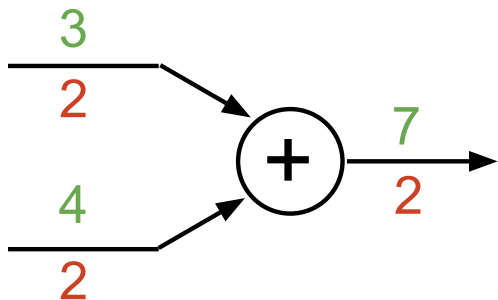


copy gate: gradient adder

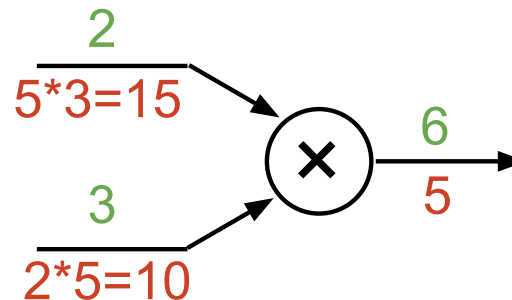


Patterns in gradient flow

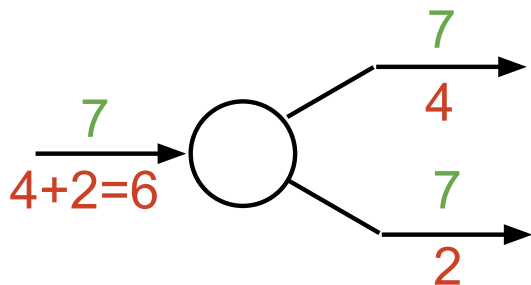
add gate: gradient distributor



mul gate: “swap multiplier”



copy gate: gradient adder



max gate: gradient router

