

Глубокое обучение в компьютерном зрении

Занятие 3 Backpropagation и Нейронные сети

Дмитрий Яшунин, к.ф.-м.н
IntelliVision

e-mail: yashuninda@yandex.ru

Update: Задания

Практика

13 заданий на 7 баллов
(7/13 ≈ 0.54 за задание)

1. knn classifier
2. SVM classifier
3. Softmax classifier
4. Two-layer neural network
5. Image features neural network
6. Fully-connected neural network
7. Batch Normalization
8. Dropout
9. Convolutional Networks
10. TensorFlow on CIFAR-10
~~Tensorflow detection models (task for a group)~~
11. Image Captioning with Vanilla RNNs
12. Image Captioning with LSTMs
~~Network Visualization: Saliency maps, Class Visualization, and Fooling Images~~
~~Style Transfer~~
13. Generative Adversarial Networks

На прошлом занятии: Классификация изображений

Ключевая задача компьютерного зрения



К какому классу принадлежит изображение?
классы: человек, животное, автомобиль ...



КОТ

На прошлом занятии: Линейный классификатор

Image



$$s = f(x, W) = Wx + b$$

Diagram illustrating the linear classification equation $s = f(x, W) = Wx + b$ with dimensions:

- s (scores): 10x1 (green box)
- $f(x, W)$ (function): 10x3072 (red box)
- W (weights or parameters): 10x3072 (red box)
- x (image pixels): 3072x1 (blue box)
- b (bias): 10x1 (purple box)

s – scores

W – weights or parameters

x – image pixels

b – bias

Array of **32x32x3** numbers
(3072 numbers total)

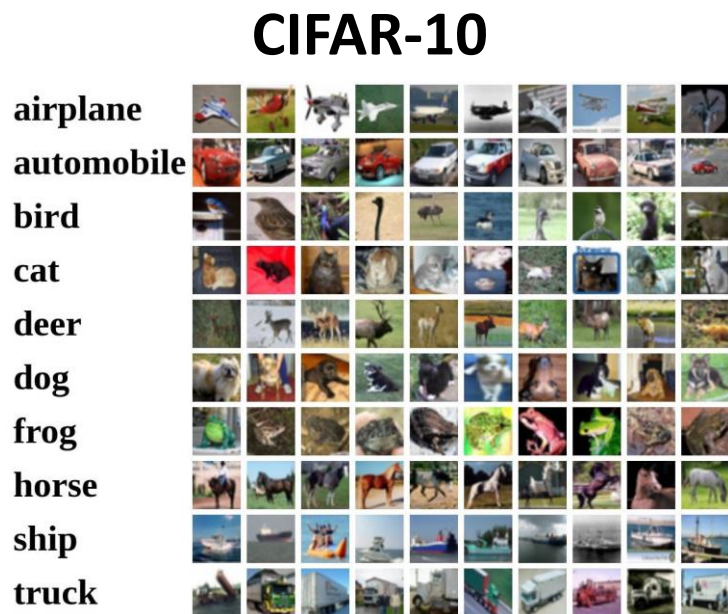
CIFAR-10

50,000 training images

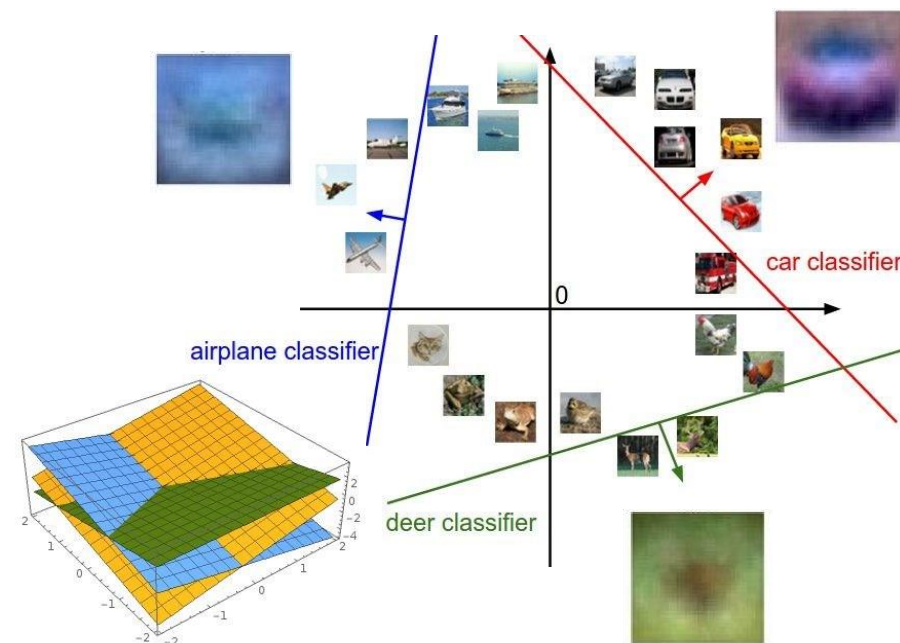
10,000 testing images

10 classes

На прошлом занятии: Интерпретация линейного классификатора



$$f(x, W) = Wx + b$$



Example trained weights of a linear classifier trained on CIFAR-10:



На прошлом занятии: Функции ошибки

Image



x_i - image

y_i - label, element of a set $\{0, 1, \dots\}$

scores $s = f(x_i, W) = [s_0, \dots, s_{y_i}, \dots]$

Loss over dataset:

$$L = \frac{1}{N} \sum_{i=1}^N L_i$$

Multiclass SVM (hinge) loss:

$$L_i = \sum_{i \neq y_i} \max(0, s_i - s_{y_i} + 1)$$

Cross-entropy (softmax) loss:

$$L_i = -\log \frac{e^{s_{y_i}}}{\sum_j e^{s_j}}$$

На прошлом занятии: Регуляризация

Softmax or
SVM

Full loss

$$L = \underbrace{\frac{1}{N} \sum_{i=1}^N L_i}_{\text{Data loss}} + \underbrace{\lambda R(W)}_{\text{Regularization}}$$

λ - regularization strength (hyperparameter)

How do we find the best W ?

L2 regularization

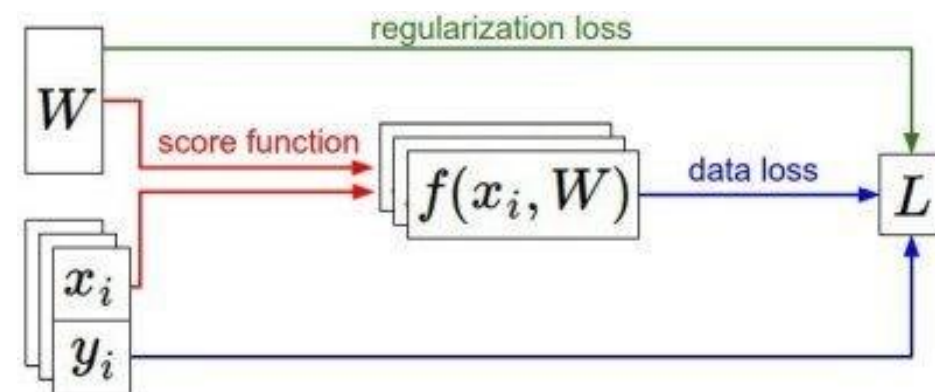
$$R(W) = \sum_k \sum_l W_{k,l}^2$$

L1 regularization

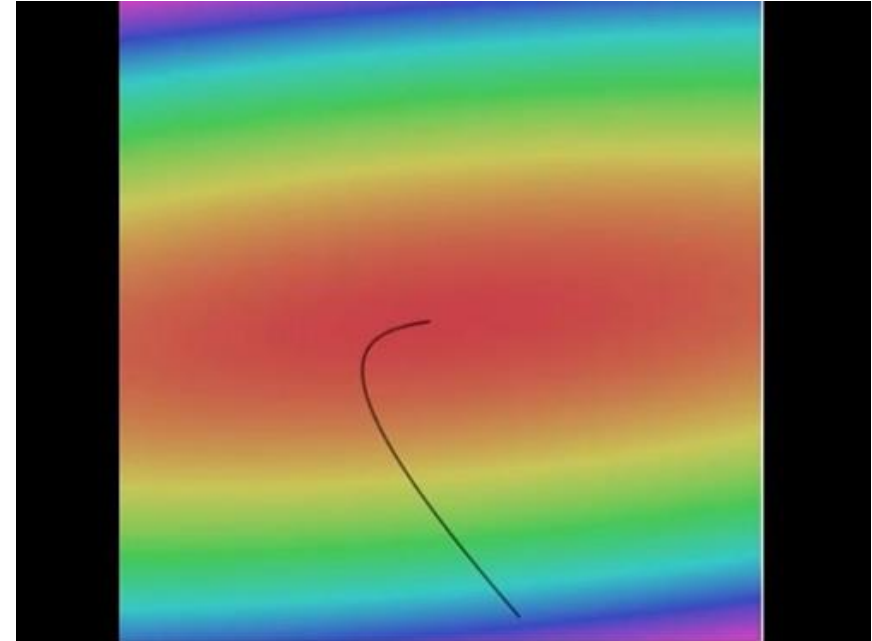
$$R(W) = \sum_k \sum_l |W_{k,l}|$$

Elastic net (L1 + L2)

$$R(W) = \sum_k \sum_l \beta W_{k,l}^2 + |W_{k,l}|$$



Оптимизация



```
# Vanilla Minibatch Gradient Descent
```

```
while True:
```

```
    data_batch = sample_training_data(data, 256) # sample 256 examples
```

```
    weights_grad = evaluate_gradient(loss_fun, data_batch, weights)
```

```
    weights += - step_size * weights_grad # perform parameter update
```

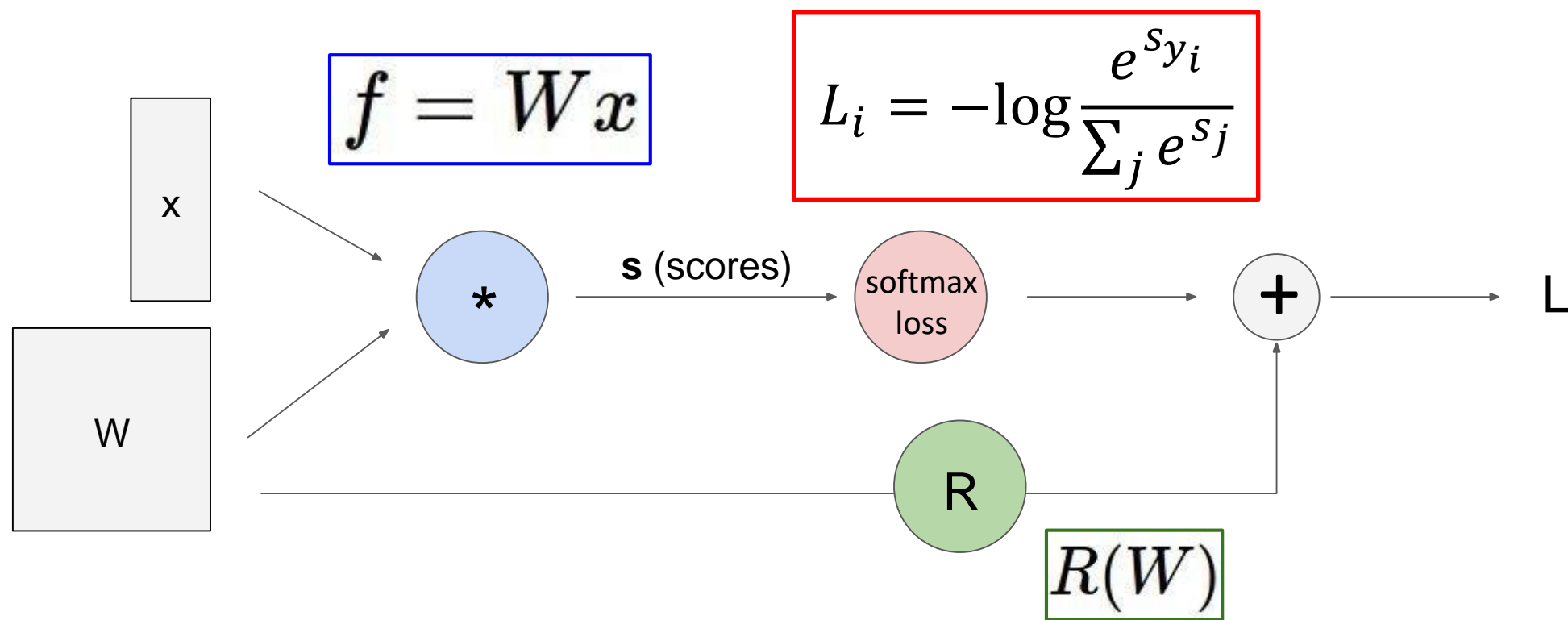

Метод градиентного спуска

$$\frac{dL(w)}{dw} = \lim_{h \rightarrow 0} \frac{L(w + h) - L(w)}{h}$$

Численные градиенты: медленно, не точно, быстро реализовать

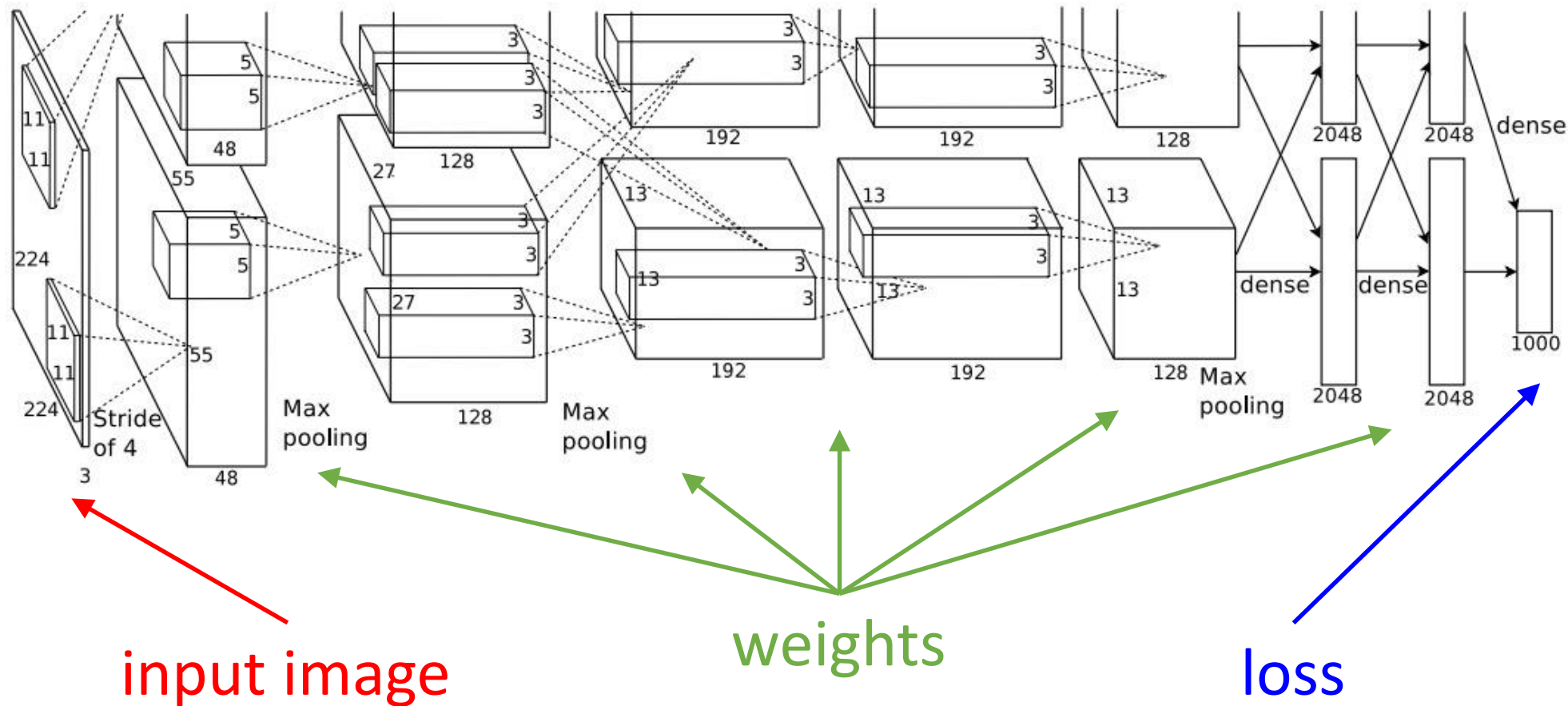
Аналитические градиенты: быстро, точно, можно ошибиться

Вычислительный граф



Вычислительный граф: Convolutional Network

AlexNet

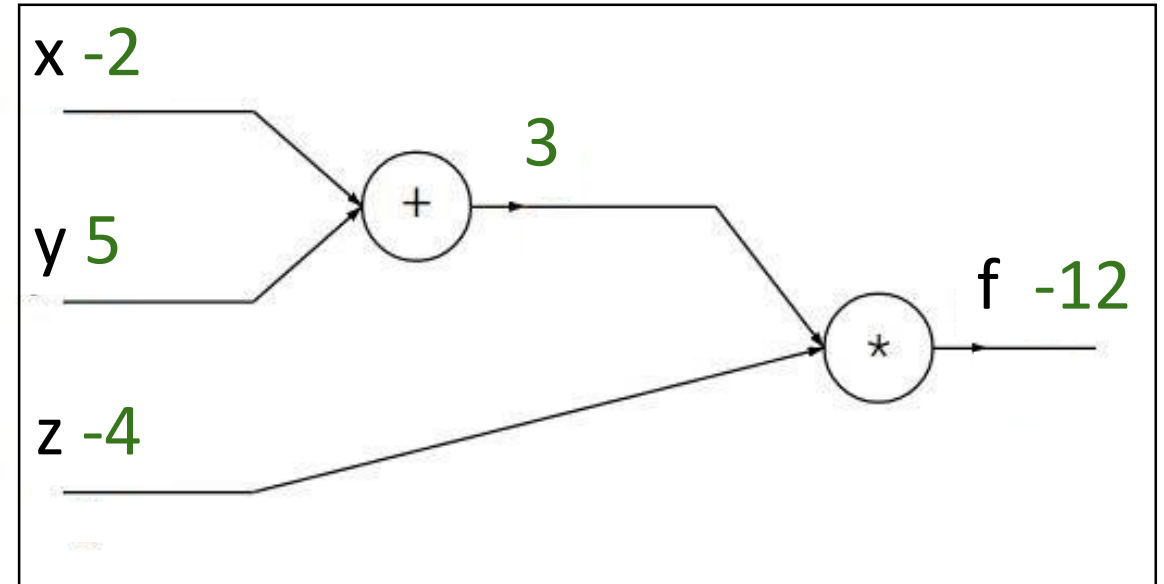


Backpropagation

Backpropagation: пример

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

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



Backpropagation: пример

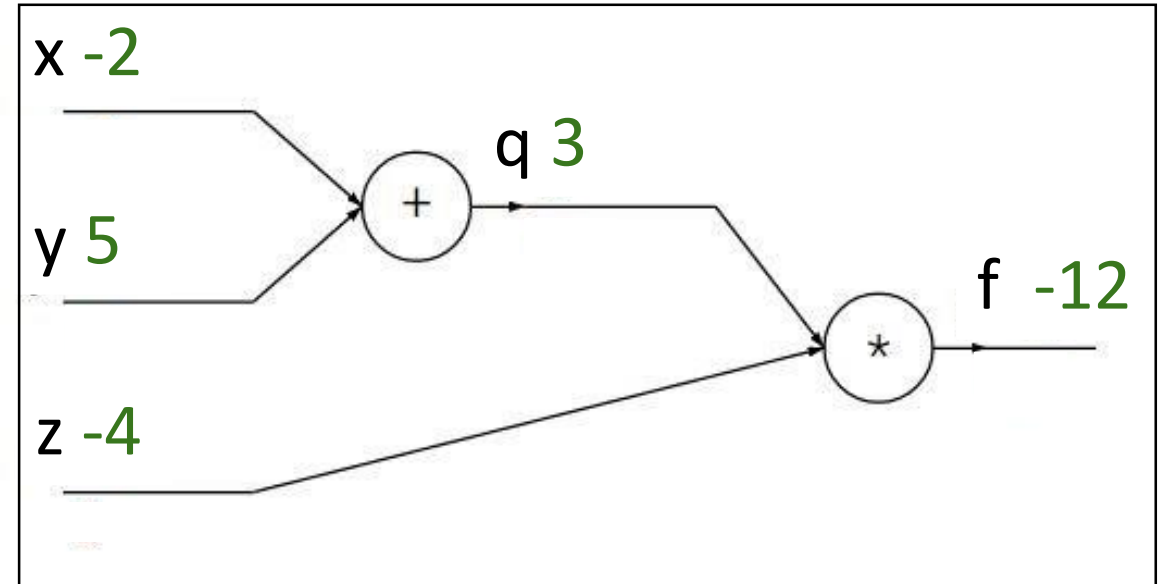
$$f(x, y, z) = (x + y)z$$

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: пример

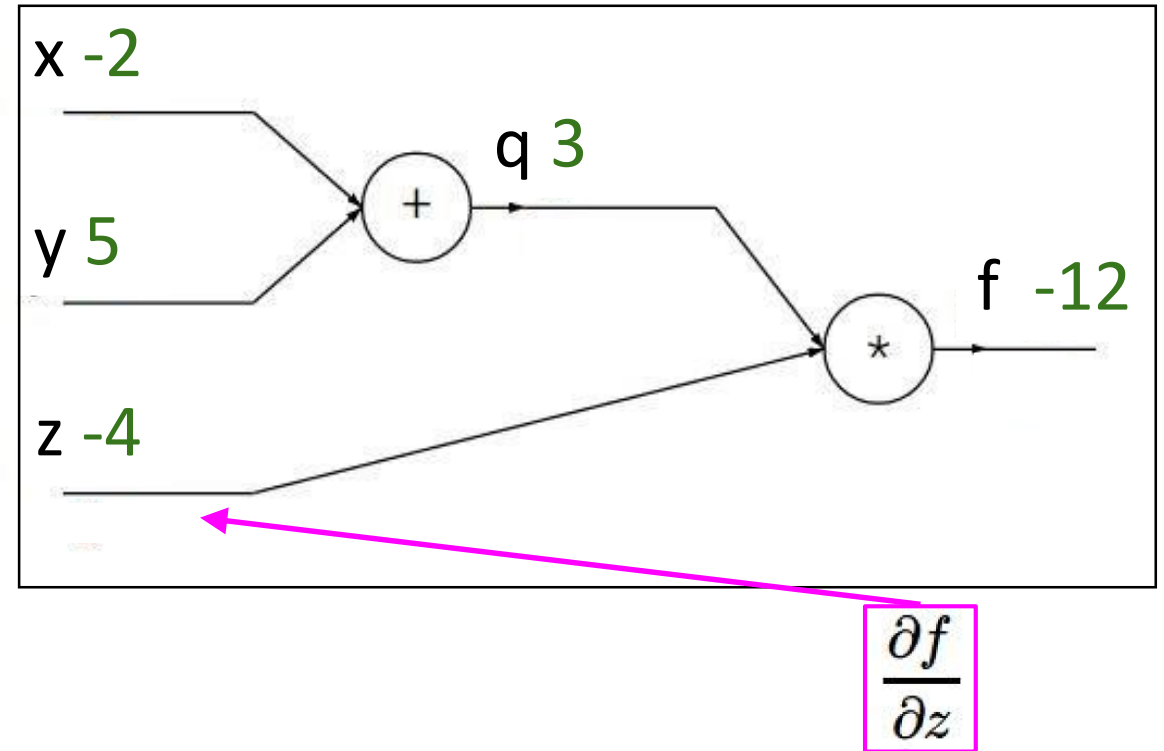
$$f(x, y, z) = (x + y)z$$

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: пример

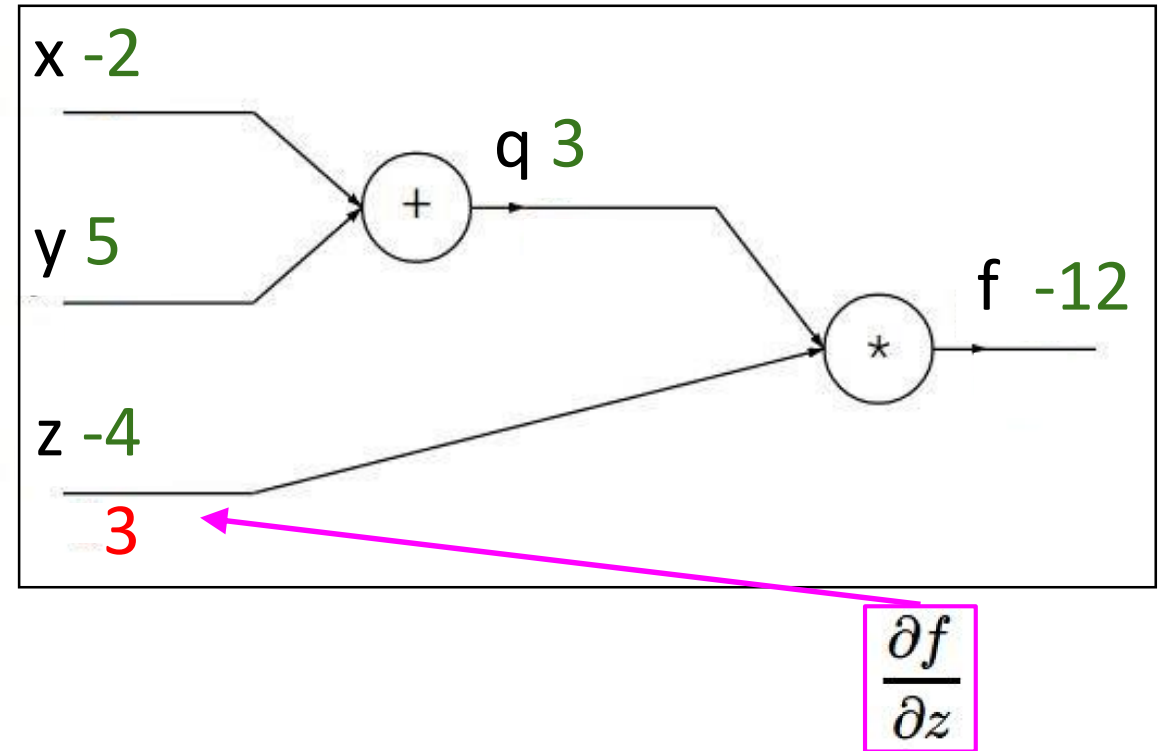
$$f(x, y, z) = (x + y)z$$

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: пример

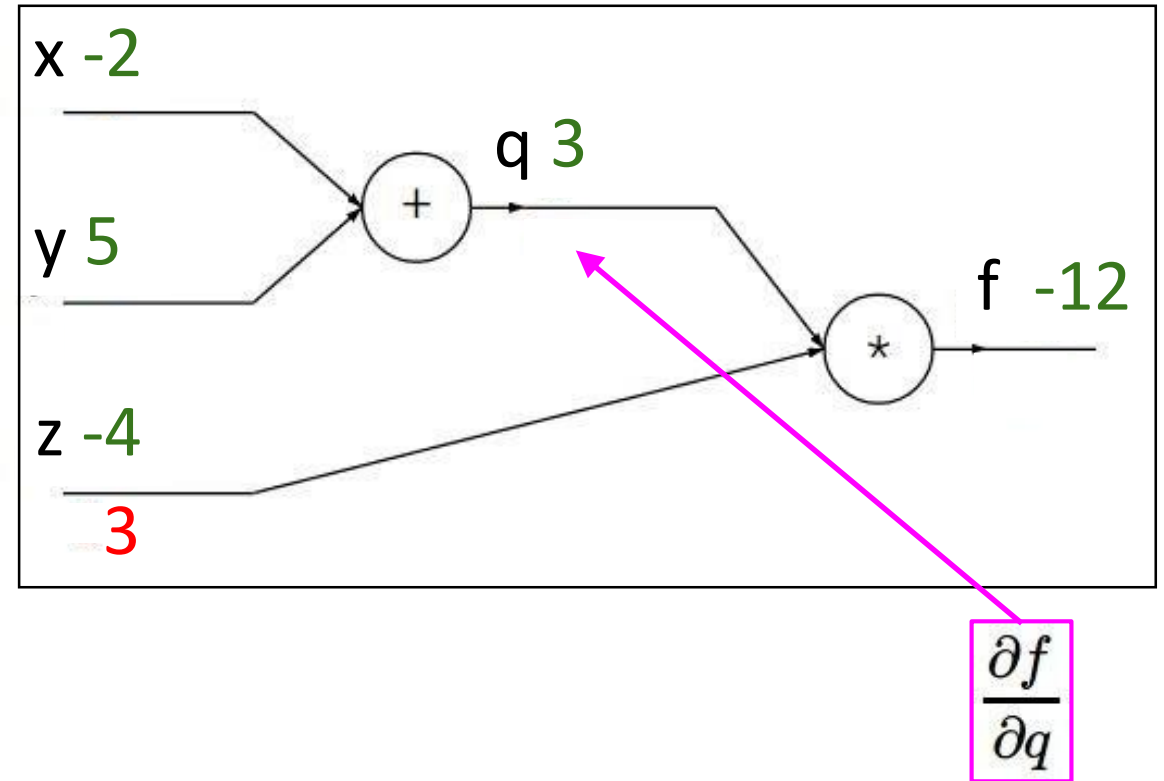
$$f(x, y, z) = (x + y)z$$

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: пример

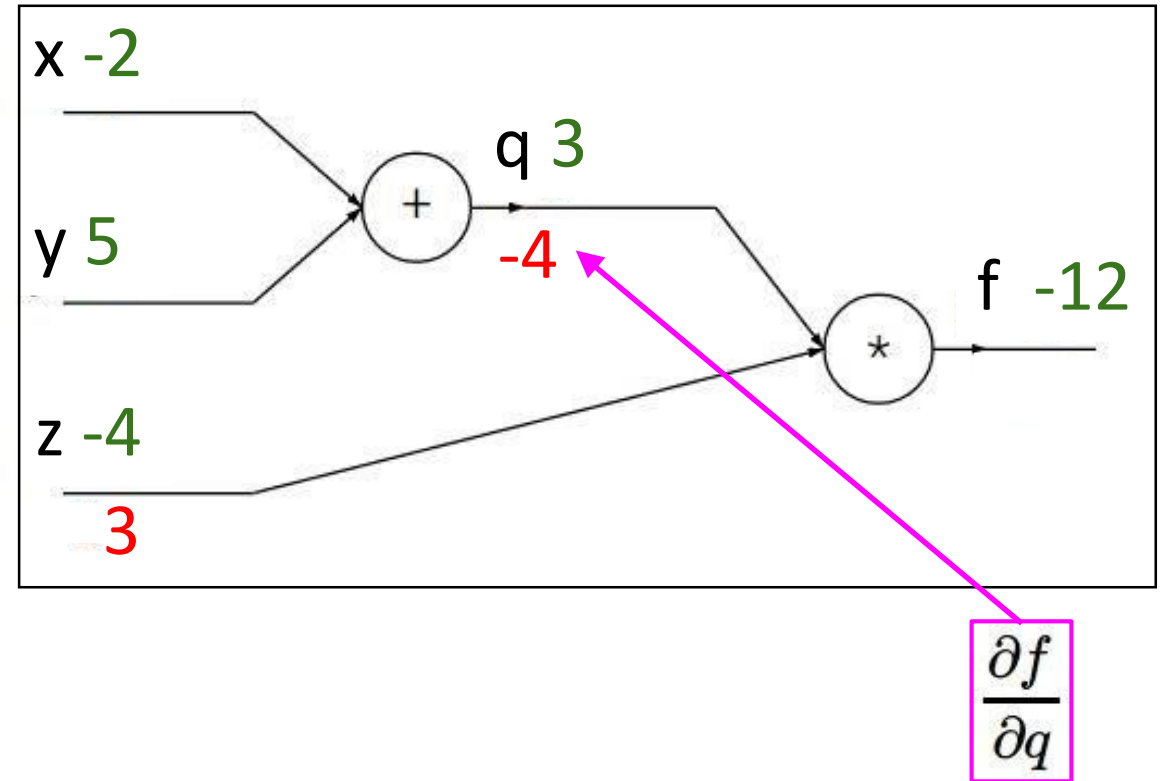
$$f(x, y, z) = (x + y)z$$

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: пример

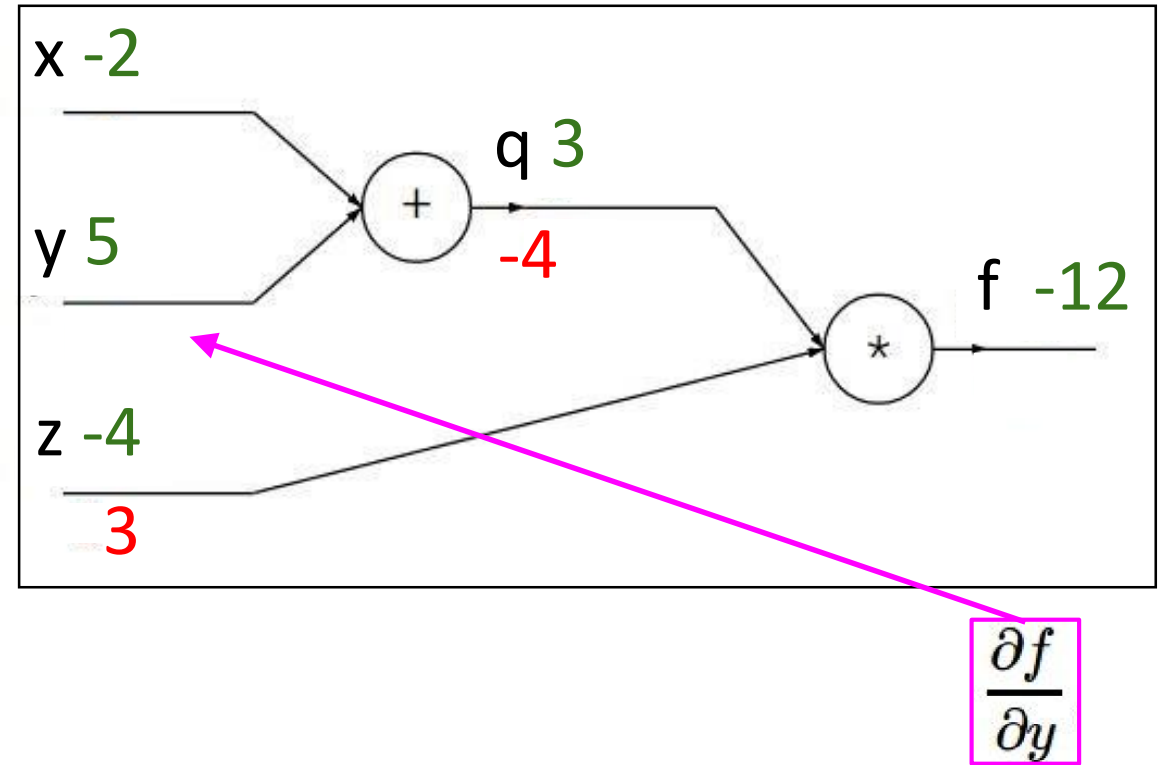
$$f(x, y, z) = (x + y)z$$

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: пример

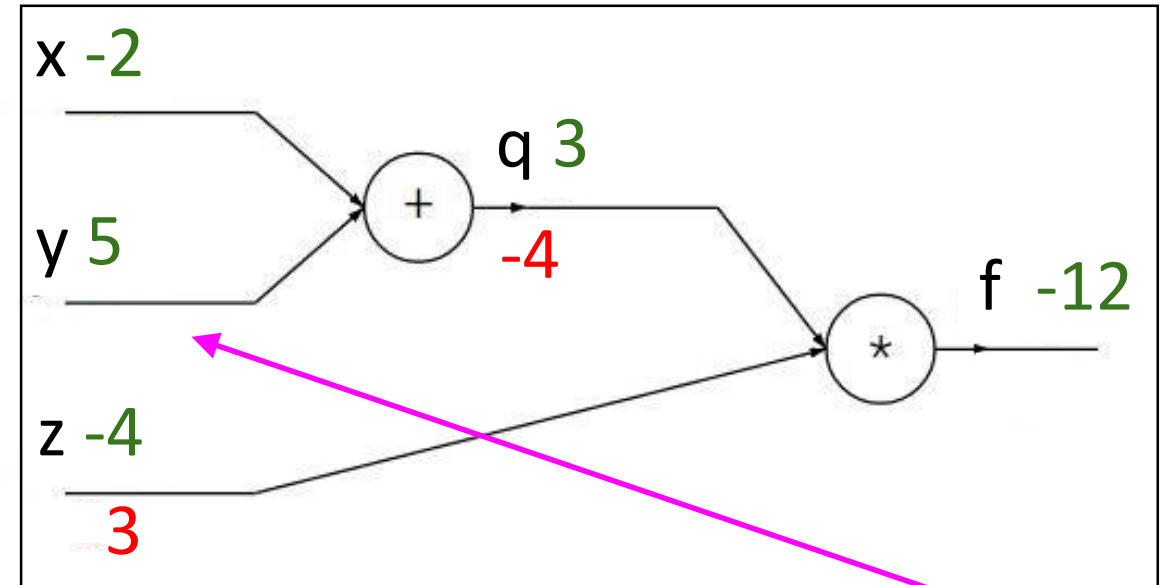
$$f(x, y, z) = (x + y)z$$

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



Правило дифференцирования:

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

Входящий
градиент

Локальный
градиент

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

Backpropagation: пример

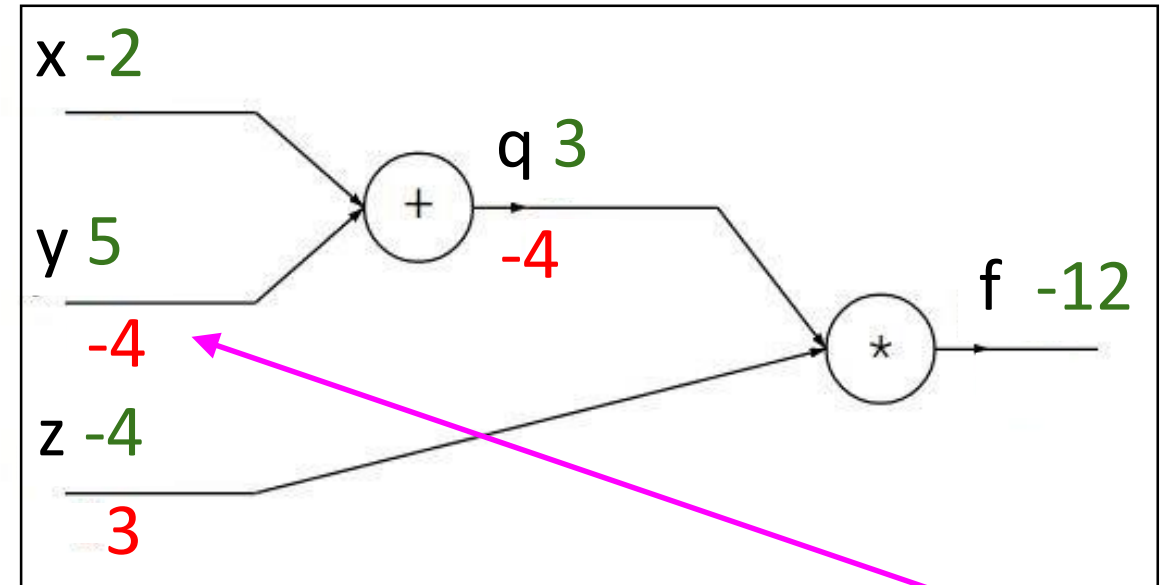
$$f(x, y, z) = (x + y)z$$

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



Правило дифференцирования:

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

Входящий
градиент

Локальный
градиент

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

Backpropagation: пример

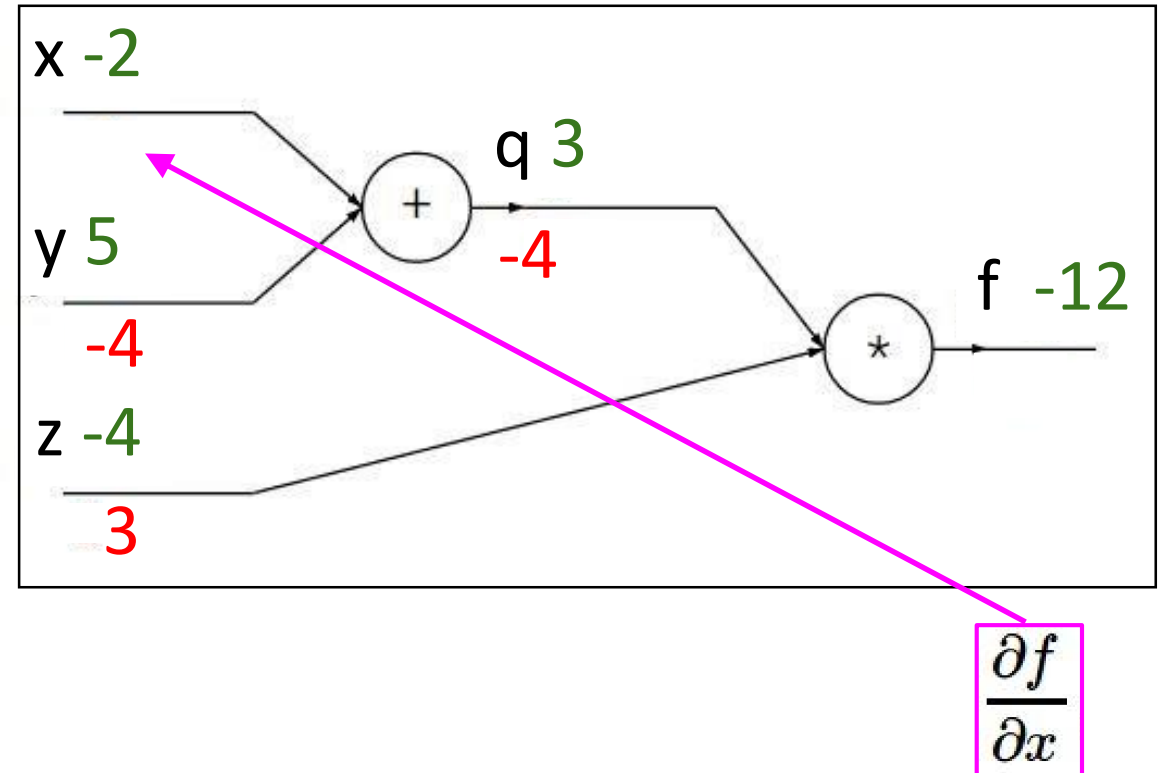
$$f(x, y, z) = (x + y)z$$

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: пример

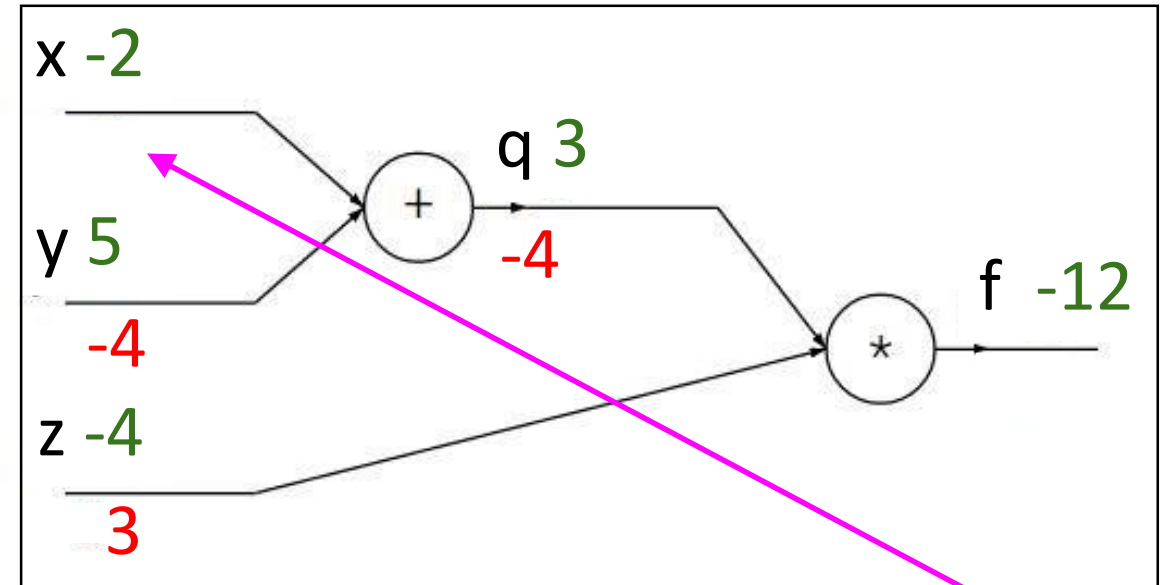
$$f(x, y, z) = (x + y)z$$

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



Правило дифференцирования:

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

Входящий
градиент

Локальный
градиент

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

Backpropagation: пример

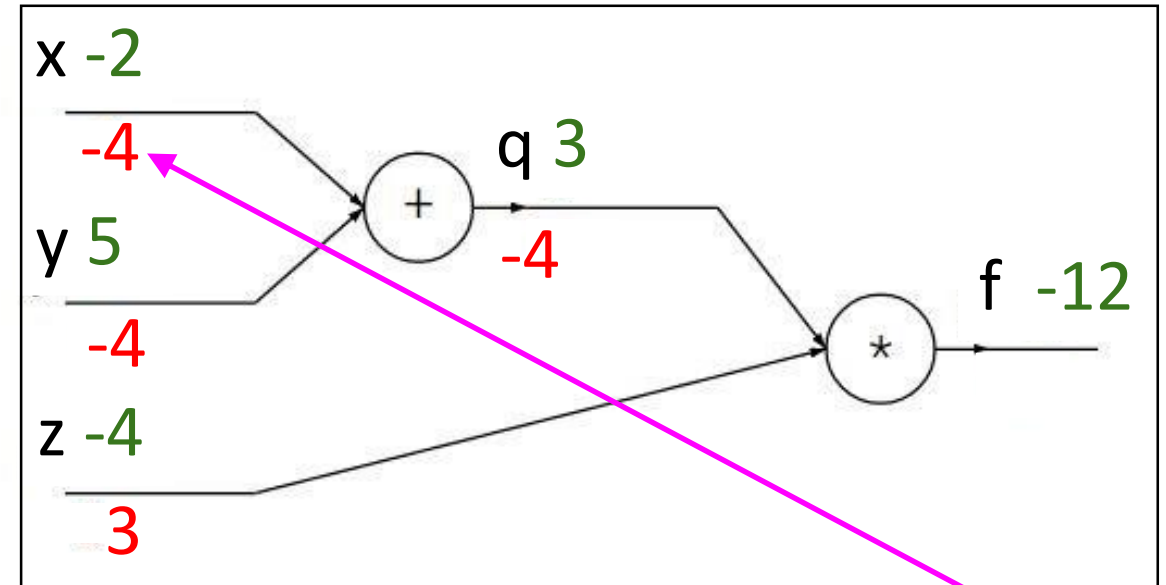
$$f(x, y, z) = (x + y)z$$

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



Правило дифференцирования:

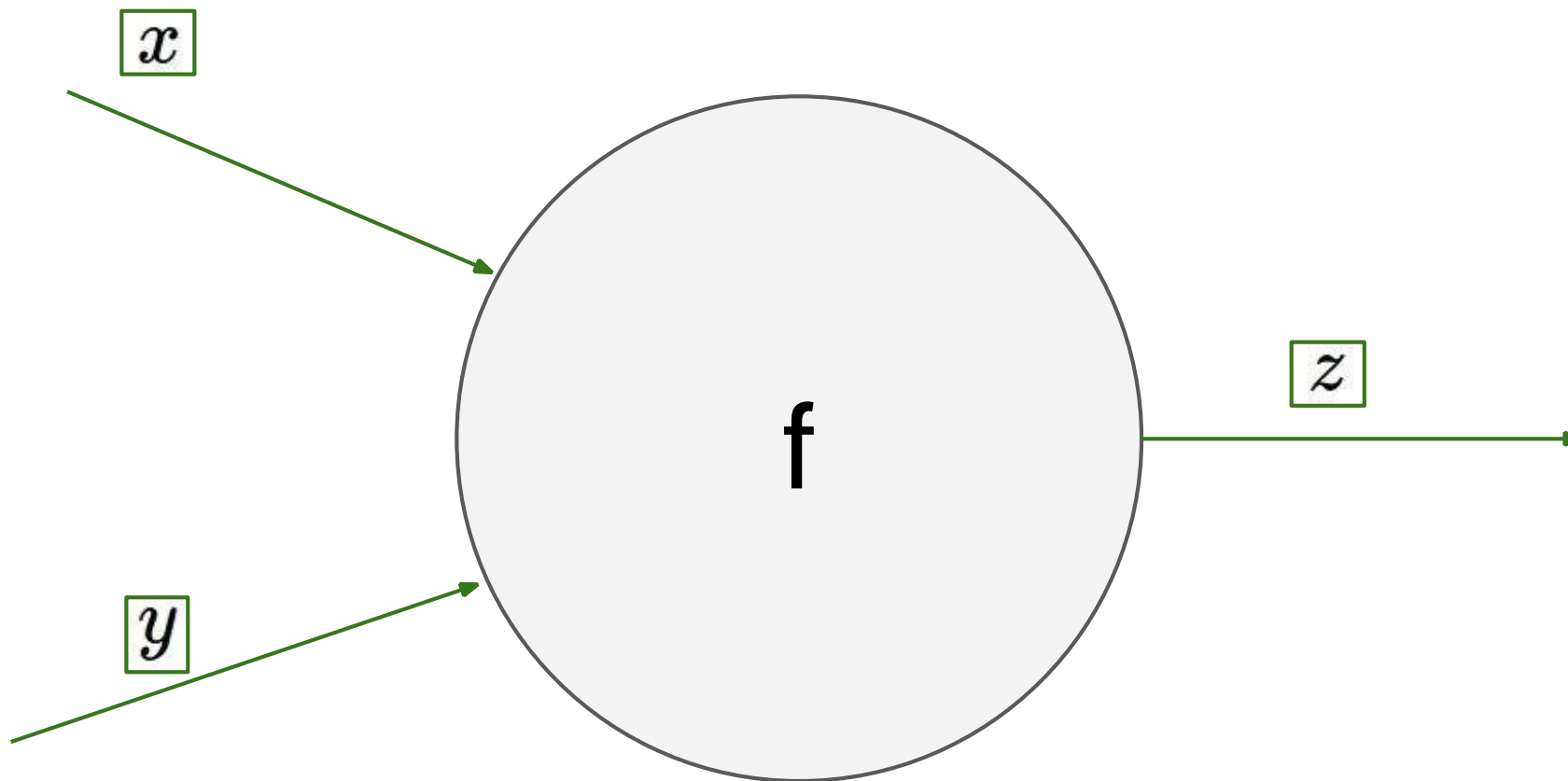
$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial x}$$

Входящий
градиент

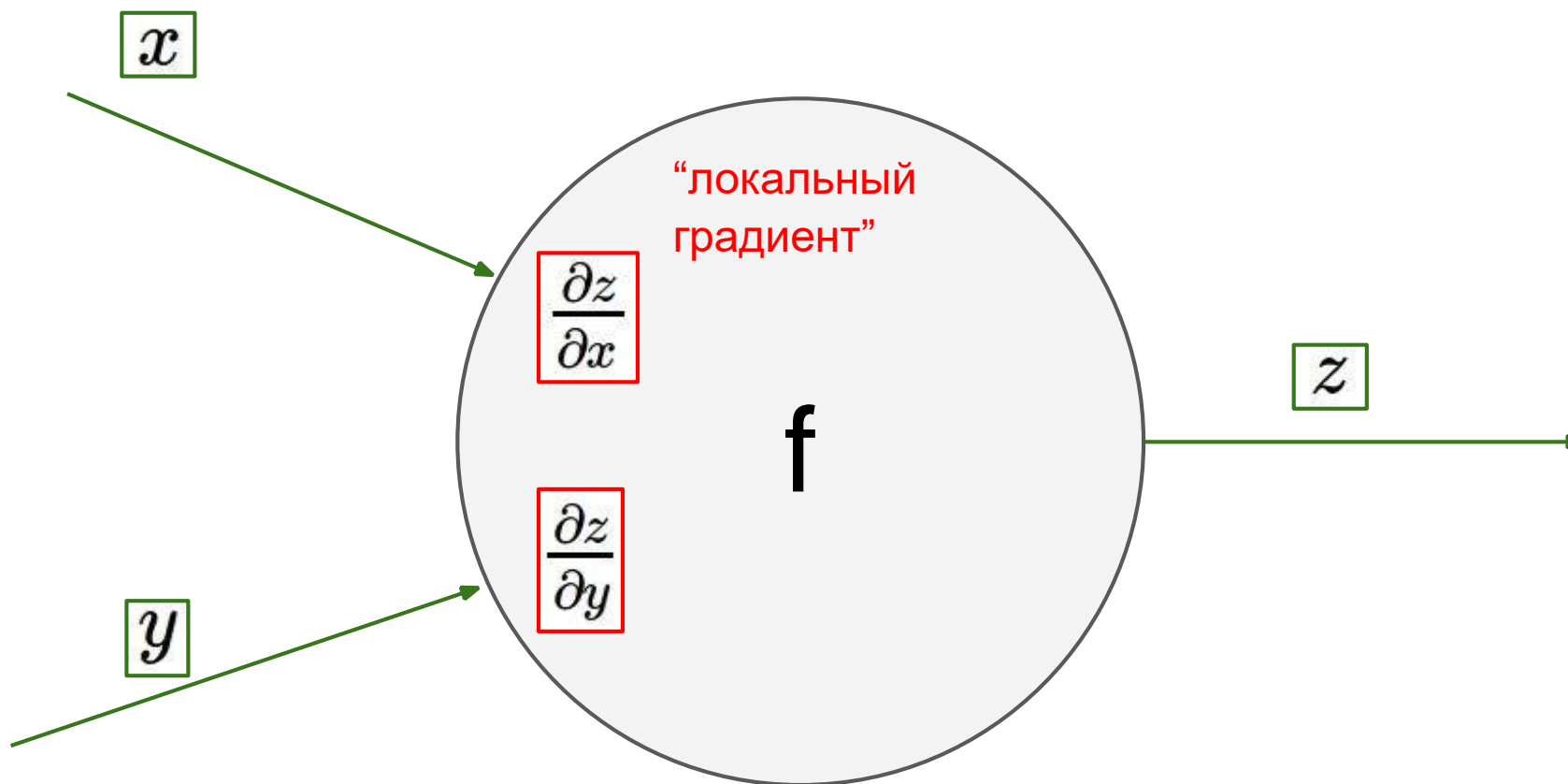
Локальный
градиент

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

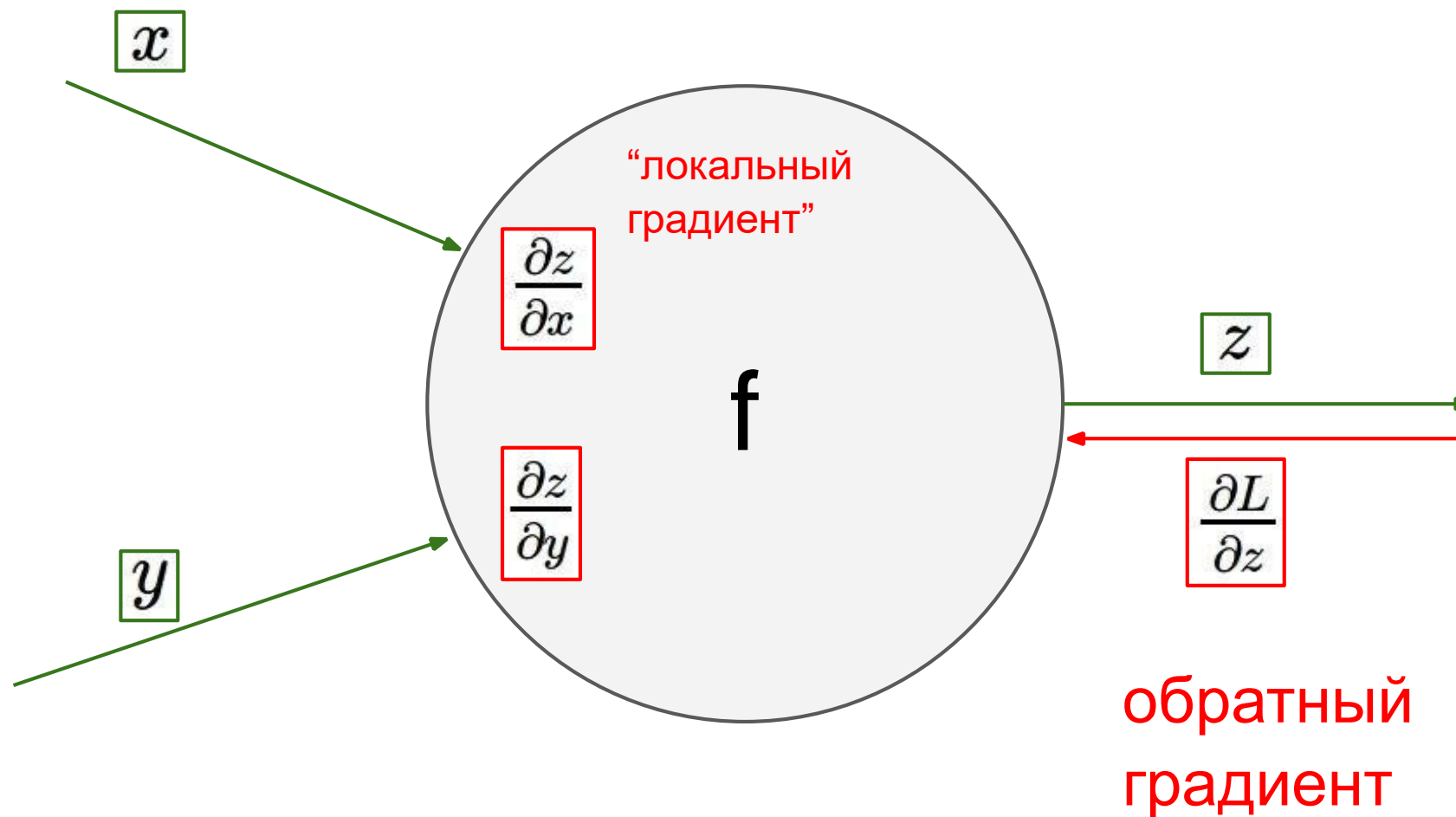
Backpropagation: общий случай



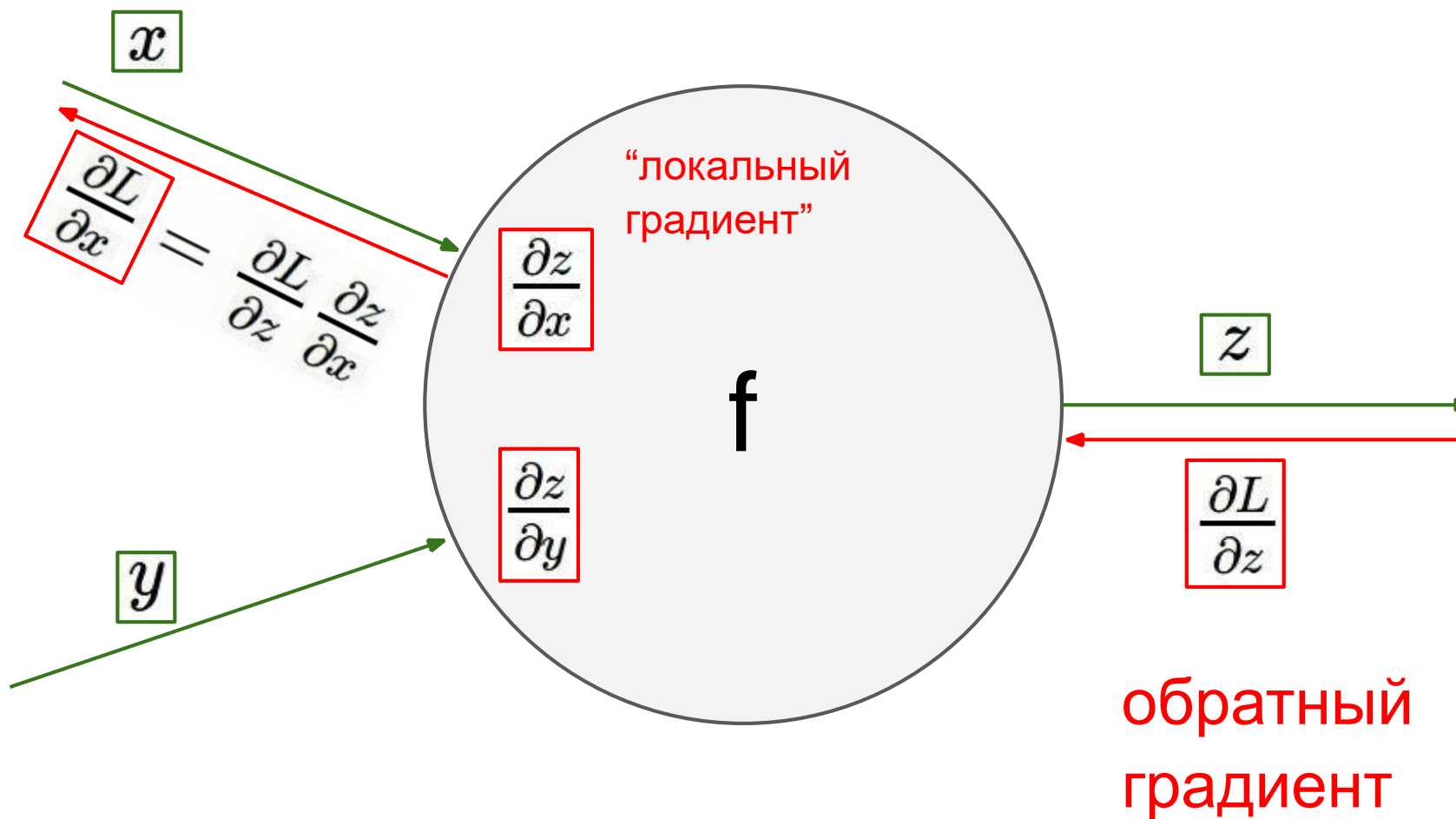
Backpropagation: общий случай



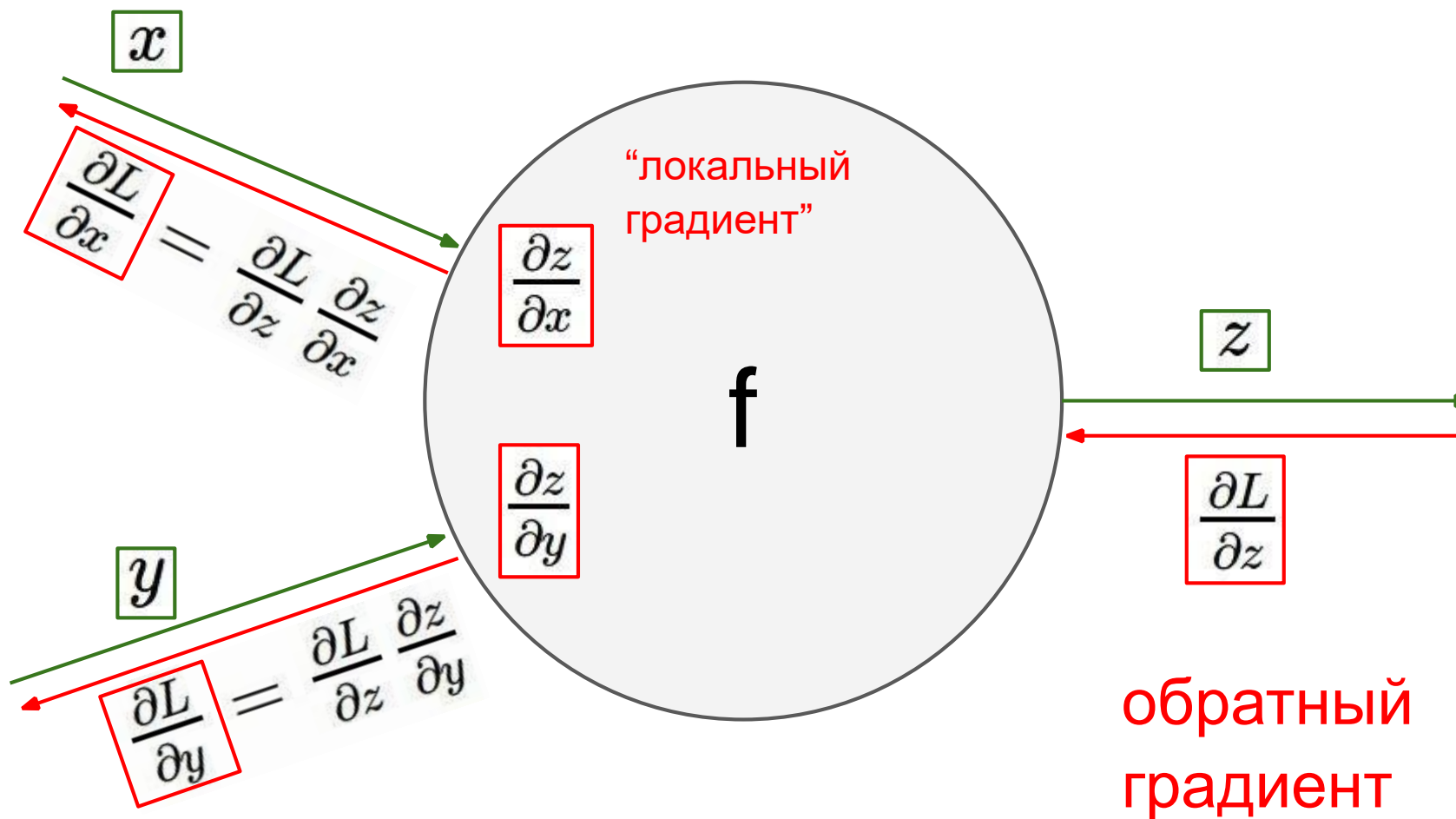
Backpropagation: общий случай



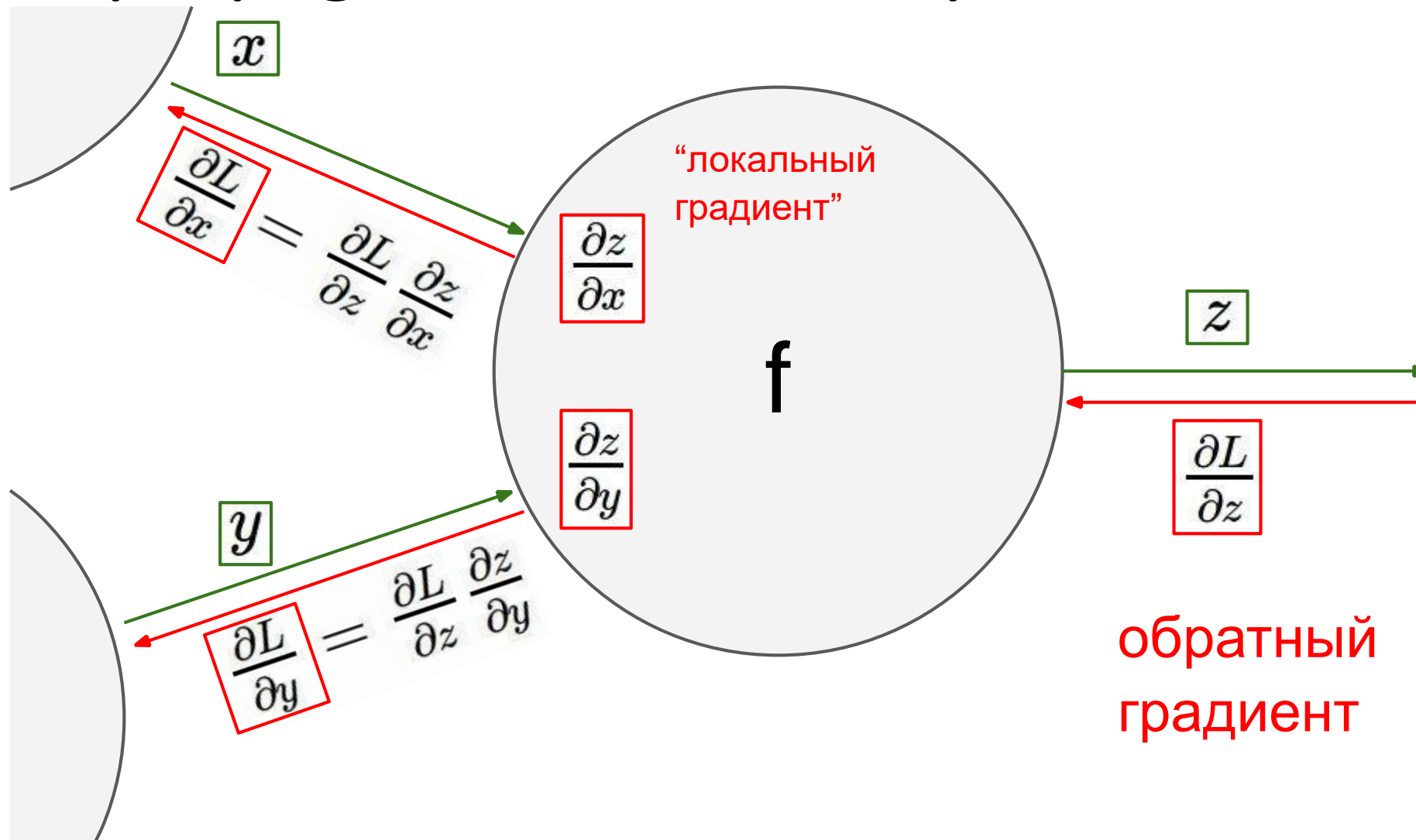
Backpropagation: общий случай



Backpropagation: общий случай



Backpropagation: общий случай



Пример:

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

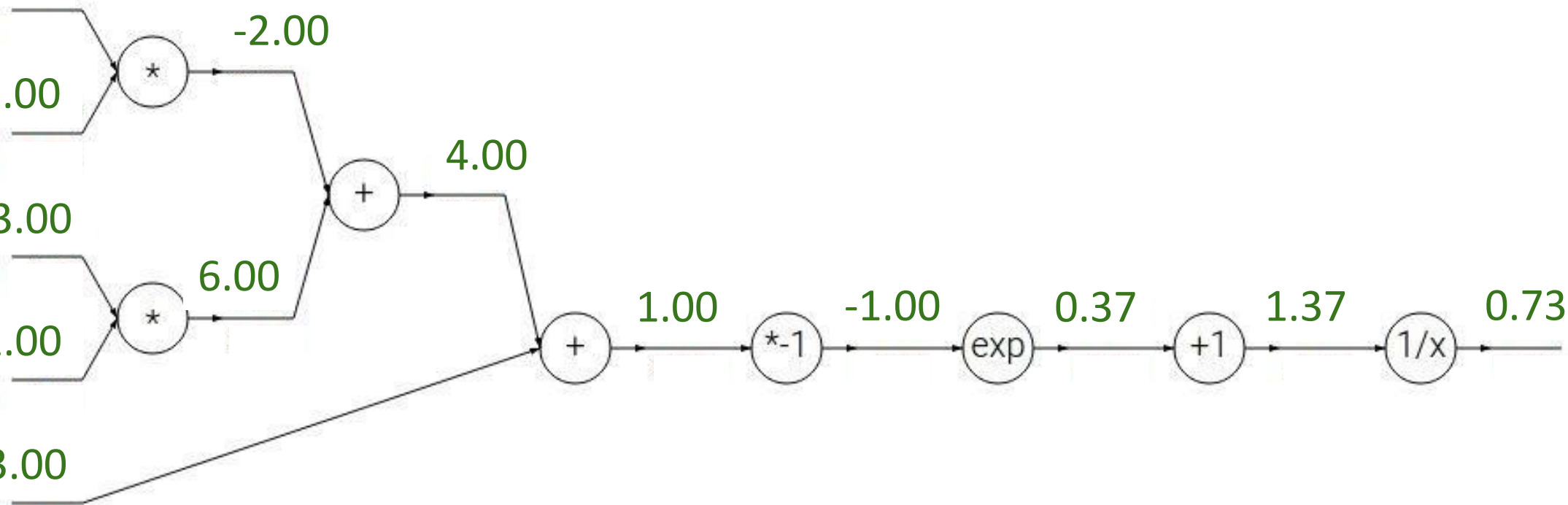
w_0 2.00

x_0 -1.00

w_1 -3.00

x_1 -2.00

w_2 -3.00



Пример:

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

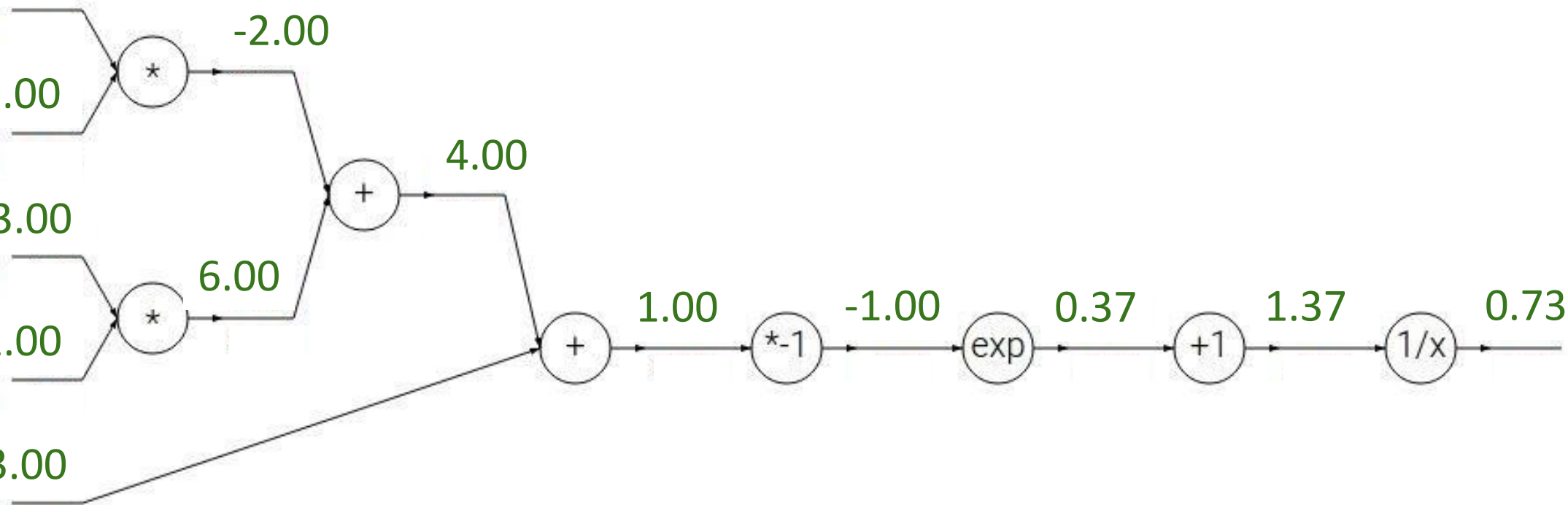
w0 2.00

x0 -1.00

w1 -3.00

x1 -2.00

w2 -3.00



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

→

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

$$f_a(x) = ax$$

→

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

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

→

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

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

→

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

Пример:

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

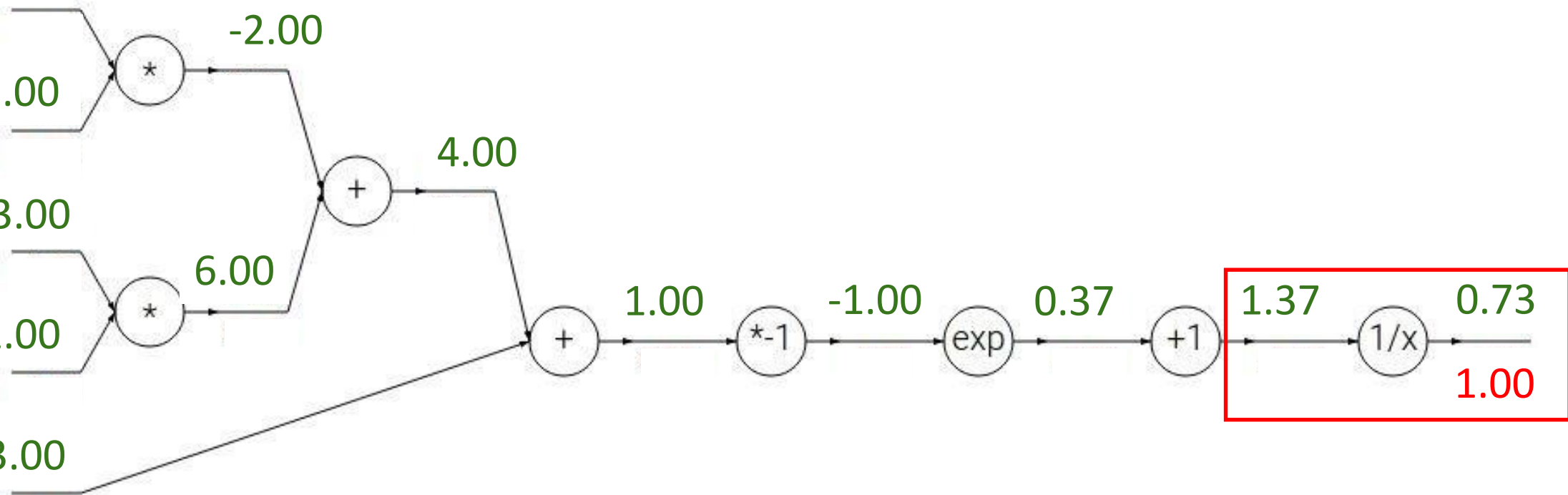
w0 2.00

x0 -1.00

w1 -3.00

x1 -2.00

w2 -3.00



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

→

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

$$f_a(x) = ax$$

→

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

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

→

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

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

→

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

Пример:

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

w0 2.00

x0 -1.00

w1 -3.00

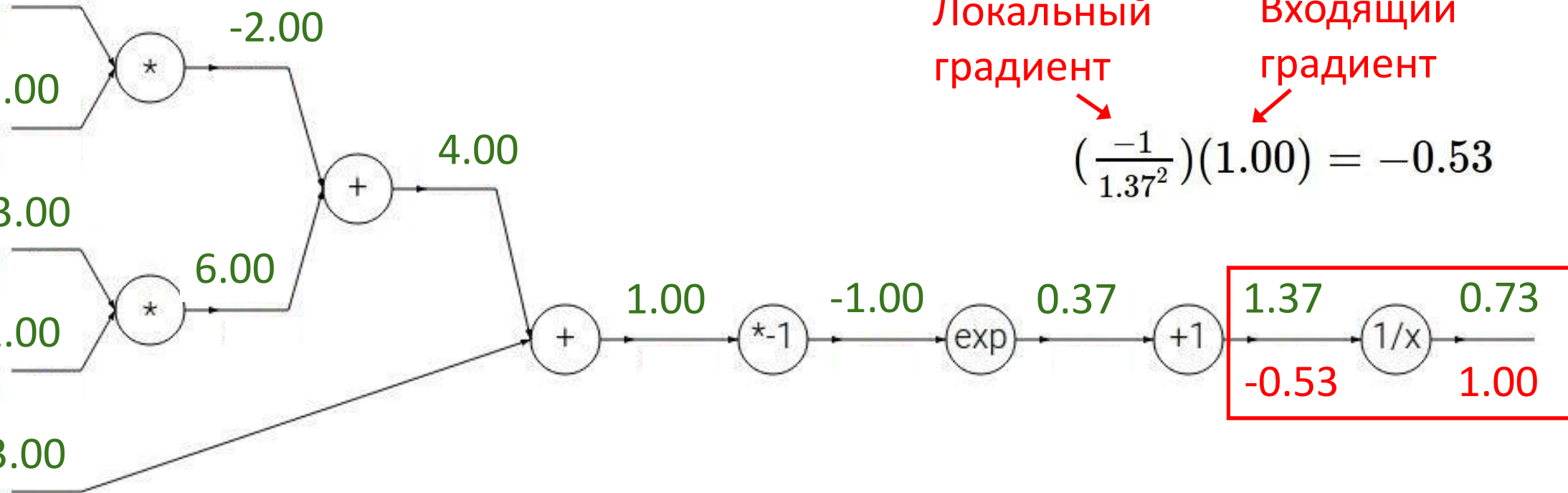
x1 -2.00

w2 -3.00

Локальный
градиент

Входящий
градиент

$$\left(\frac{-1}{1.37^2}\right)(1.00) = -0.53$$



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

→

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

$$f_a(x) = ax$$

→

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

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

→

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

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

→

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

Пример:

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

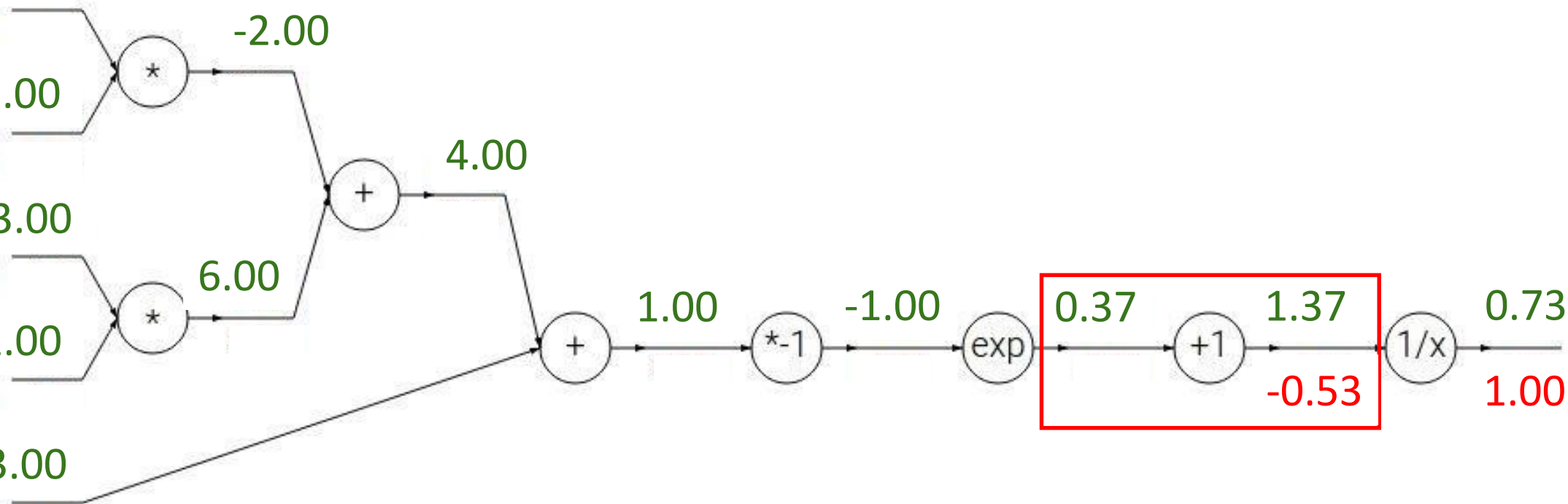
w0 2.00

x0 -1.00

w1 -3.00

x1 -2.00

w2 -3.00



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

→

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

$$f_a(x) = ax$$

→

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

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

→

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

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

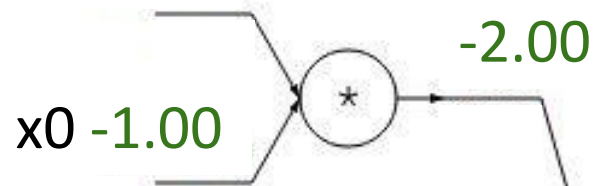
→

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

Пример:

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

w0 2.00



w1 -3.00



w2 -3.00

4.00

1.00

Локальный
градиент

Входящий
градиент

$$(1)(-0.53) = -0.53$$

-1.00

0.37

-0.53

1.37

-0.53

0.73

1.00

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

→

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

$$f_a(x) = ax$$

→

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

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

→

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

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

→

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

Пример:

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

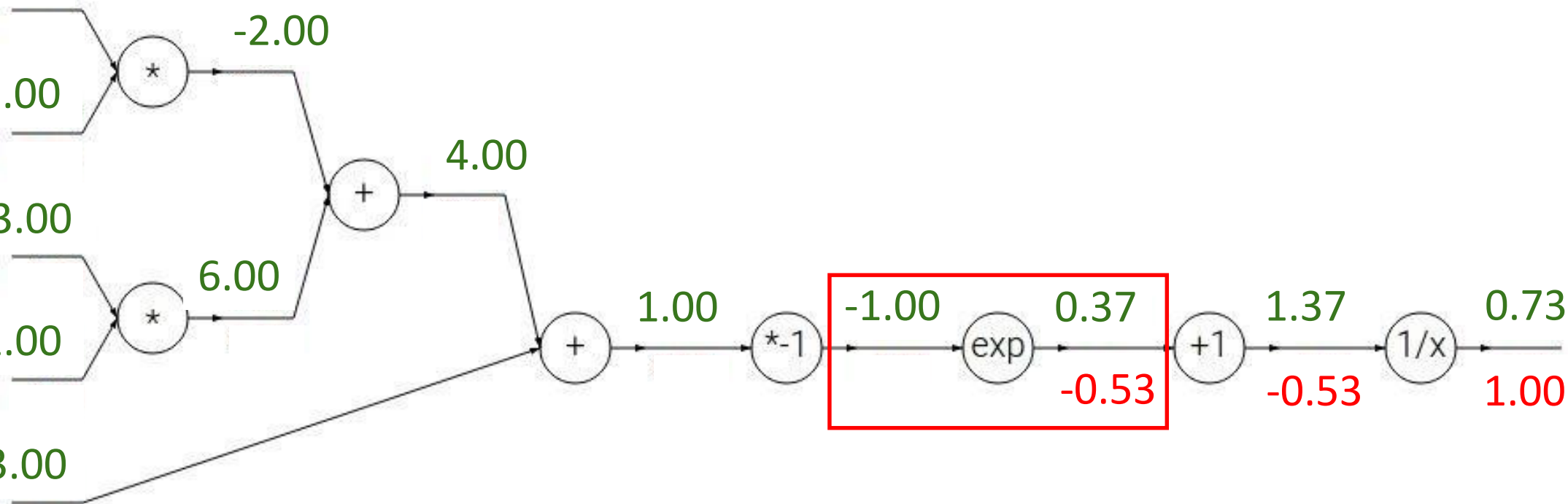
w0 2.00

x0 -1.00

w1 -3.00

x1 -2.00

w2 -3.00



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

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

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

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

→

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

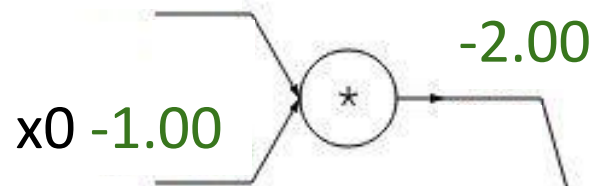
→

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

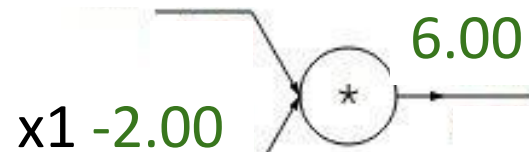
Пример:

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

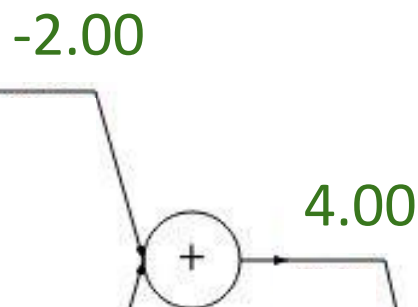
w0 2.00



w1 -3.00



w2 -3.00

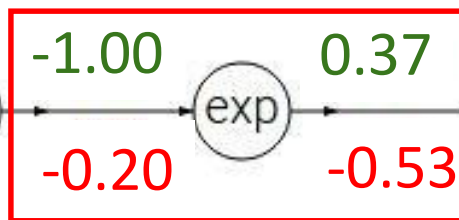


Локальный
градиент

Входящий
градиент

$$(e^{-1})(-0.53) = -0.20$$

1.00



1.37

0.73

-0.53

1.00

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

→

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

$$f_a(x) = ax$$

→

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

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

→

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

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

→

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

Пример:

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

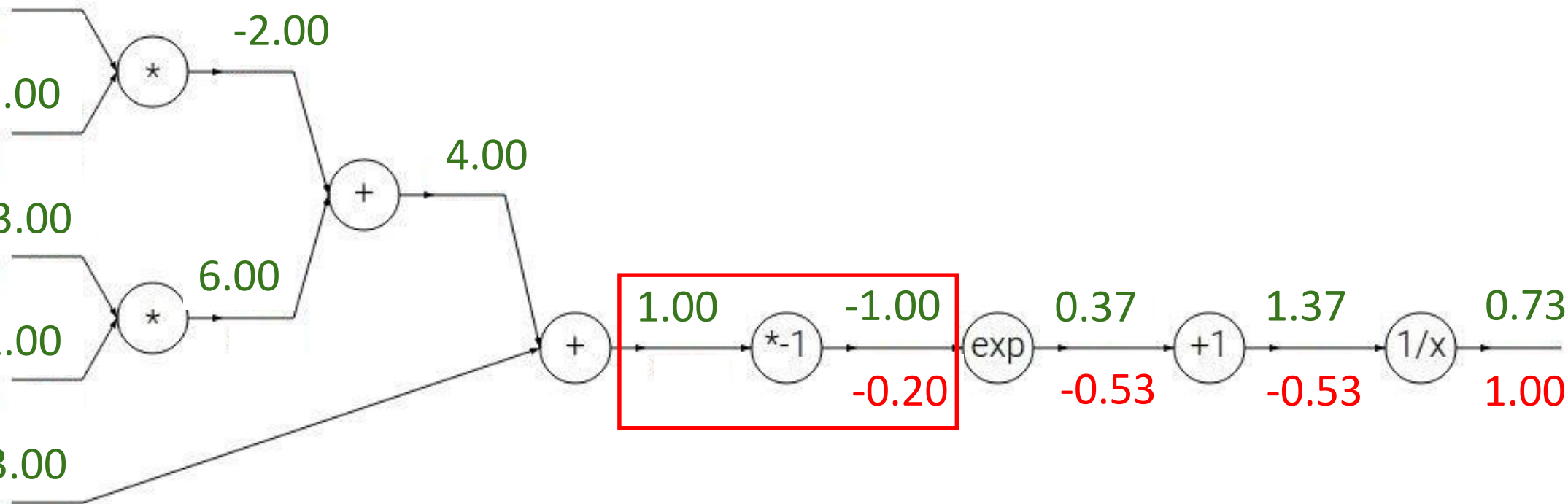
w0 2.00

x0 -1.00

w1 -3.00

x1 -2.00

w2 -3.00



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

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

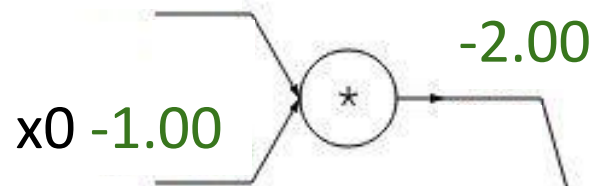
$$f(x) = \frac{1}{x} \rightarrow \frac{df}{dx} = -1/x^2$$

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

Пример:

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

w0 2.00



w1 -3.00



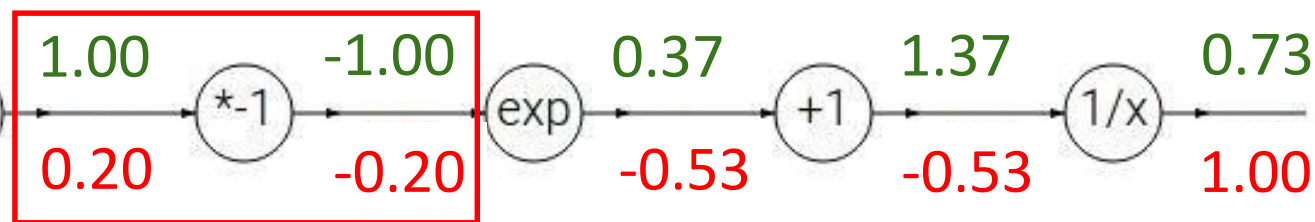
w2 -3.00

4.00

Локальный
градиент

Входящий
градиент

$$(-1) * 0.20 = 0.20$$



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

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

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

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

→

→

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

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

Пример:

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

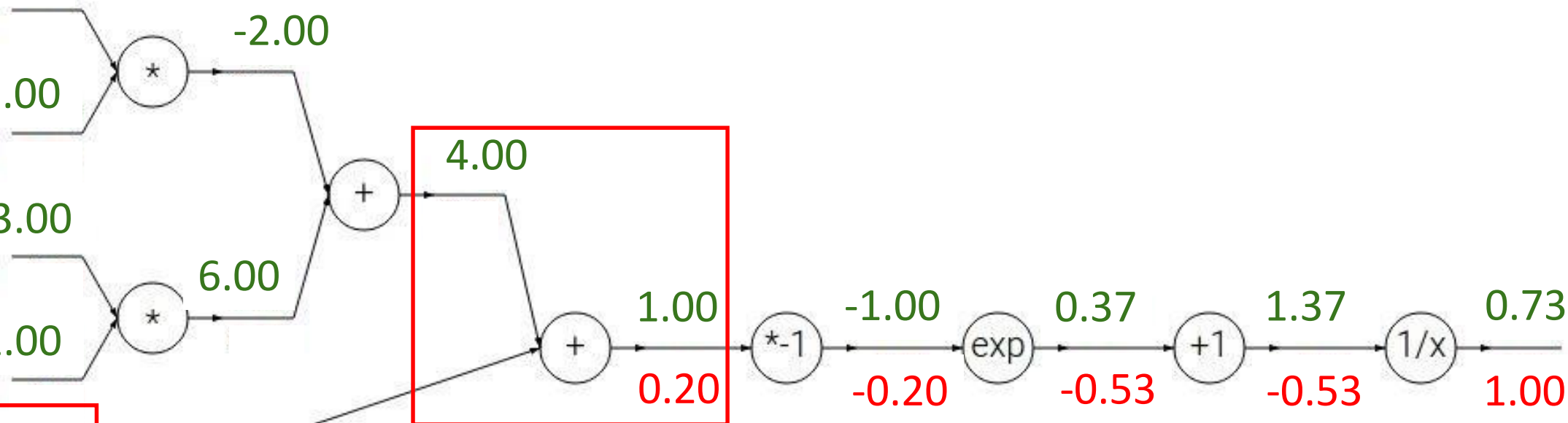
w0 2.00

x0 -1.00

w1 -3.00

x1 -2.00

w2 -3.00



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

→

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

$$f_a(x) = ax$$

→

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

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

→

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

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

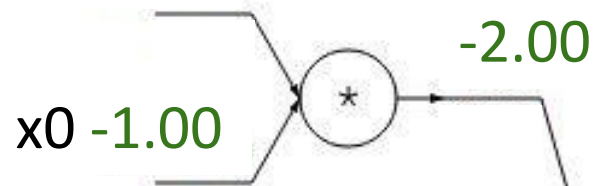
→

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

Пример:

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

w0 2.00



w1 -3.00



w2 -3.00

0.2

4.00

0.2

1.00

0.20

[локальный
градиент]

x

[входящий
градиент]

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

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

-1.00

-0.20

0.37

-0.53

1.37

-0.53

0.73

1.00

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

→

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

$$f_a(x) = ax$$

→

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

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

→

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

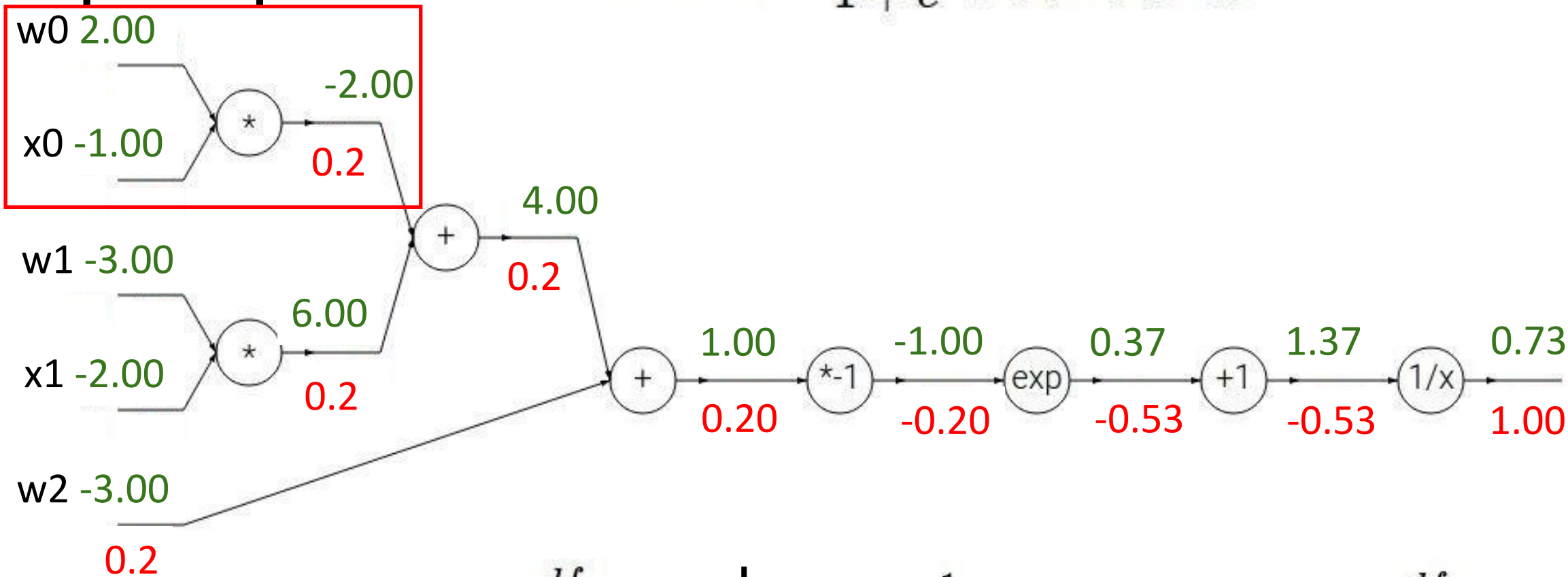
$$f_c(x) = c + x$$

→

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

Пример:

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



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

→

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

$$f_a(x) = ax$$

→

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

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

→

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

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

→

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

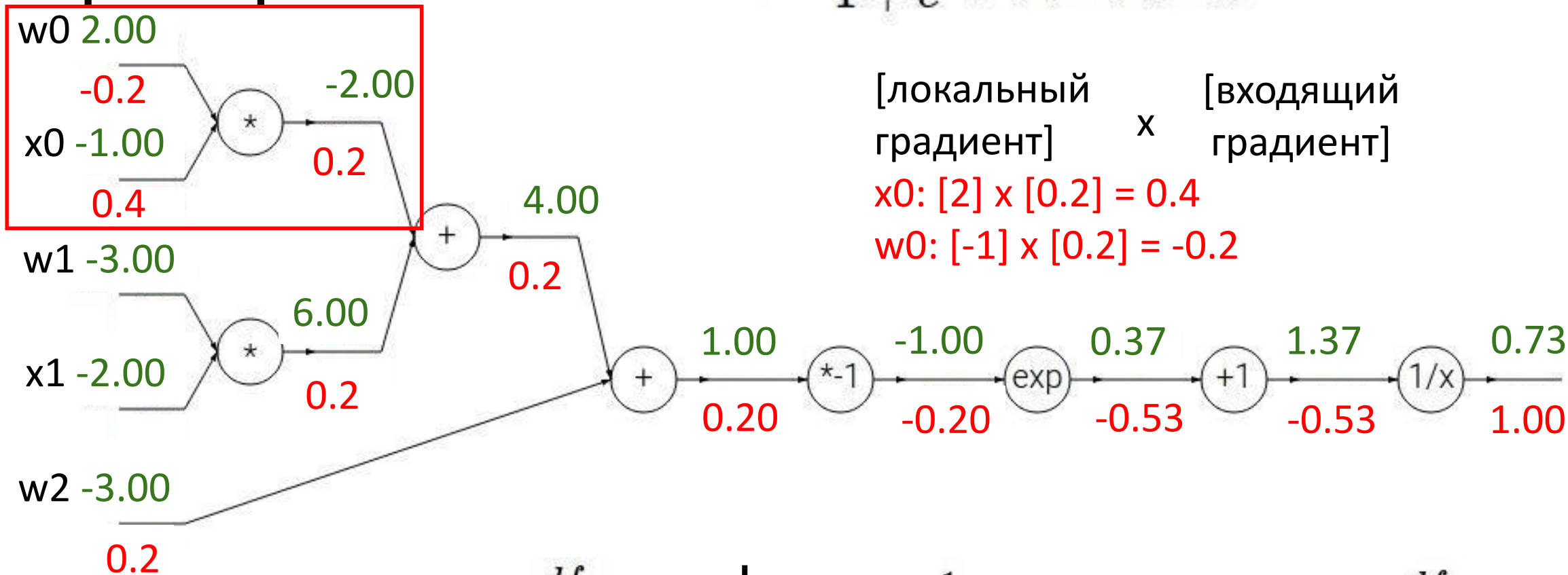
Пример:

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

[локальный
градиент] × [входящий
градиент]

$$x_0: [2] \times [0.2] = 0.4$$

$$w_0: [-1] \times [0.2] = -0.2$$



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

→

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

$$f_a(x) = ax$$

→

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

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

→

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

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

→

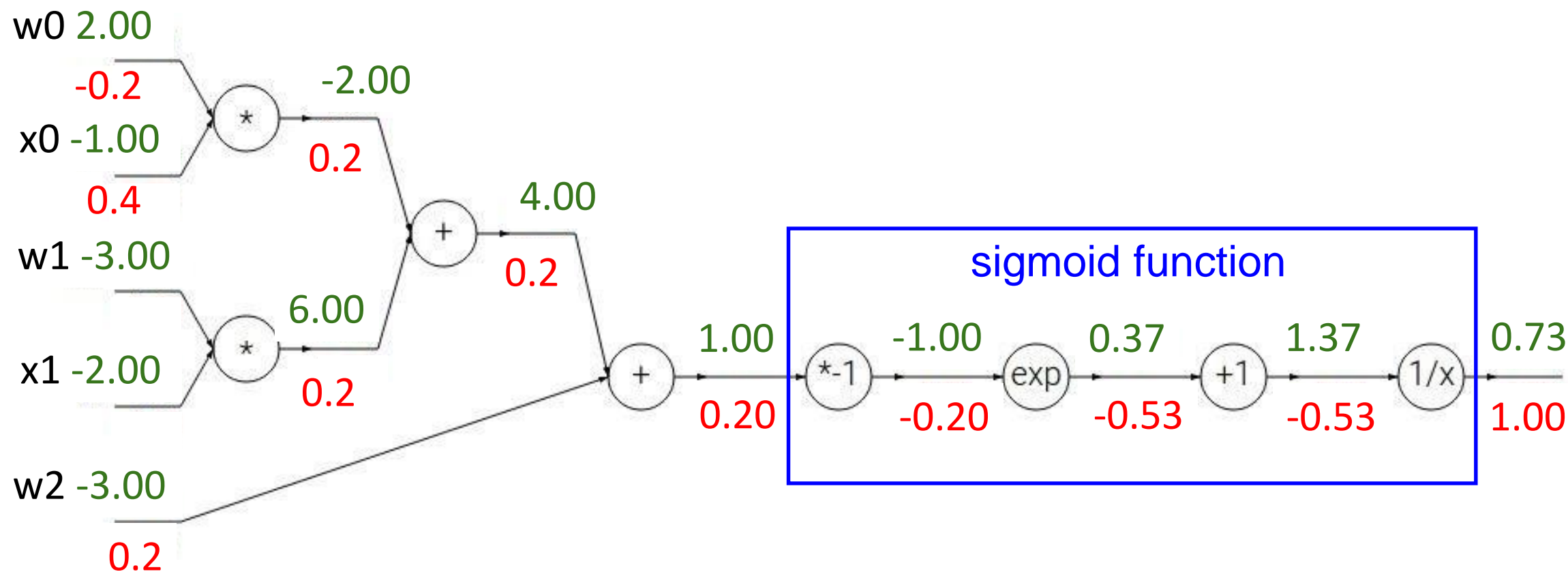
$$\frac{df}{dx} = 1$$

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

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

sigmoid function

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

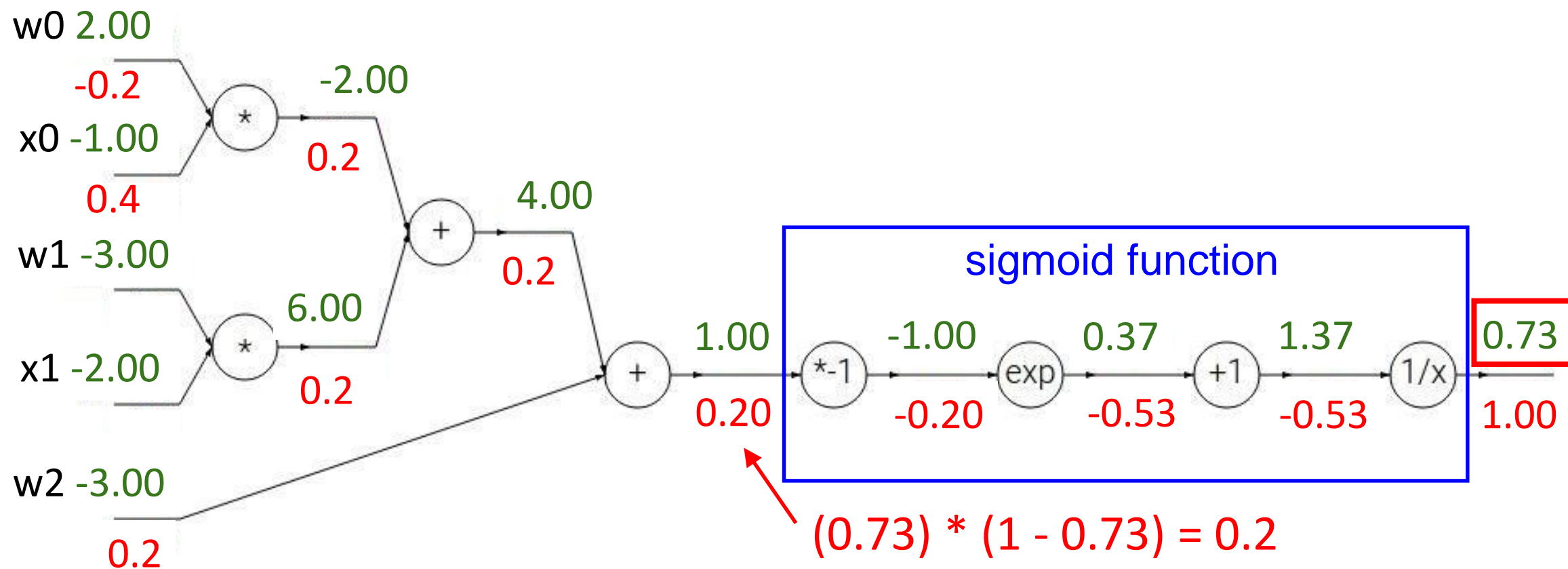


$$f(w, x) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2x_2)}}$$

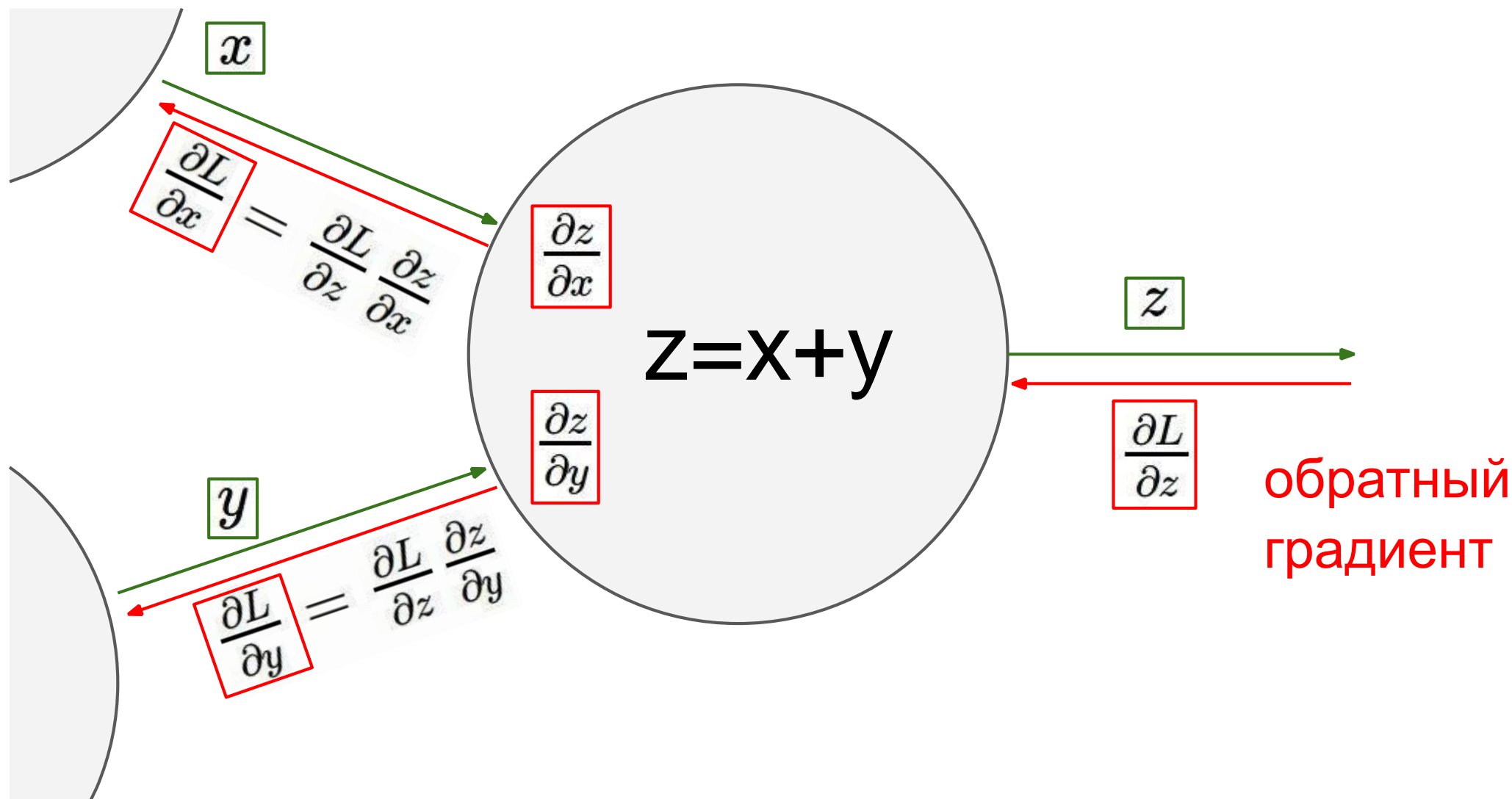
$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

sigmoid function

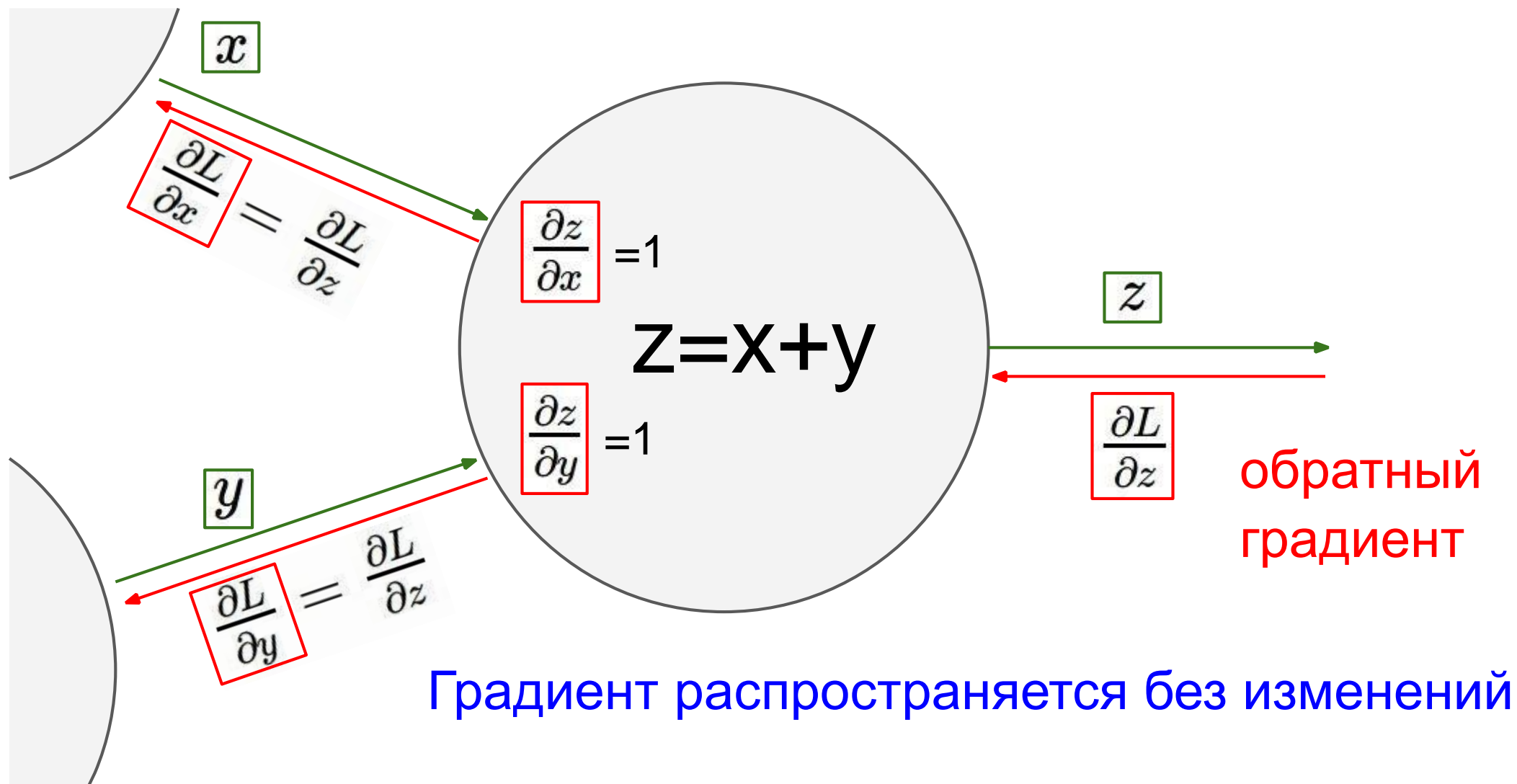
$$\frac{d\sigma(x)}{dx} = \frac{e^{-x}}{(1 + e^{-x})^2} = \left(\frac{1 + e^{-x} - 1}{1 + e^{-x}} \right) \left(\frac{1}{1 + e^{-x}} \right) = (1 - \sigma(x)) \sigma(x)$$



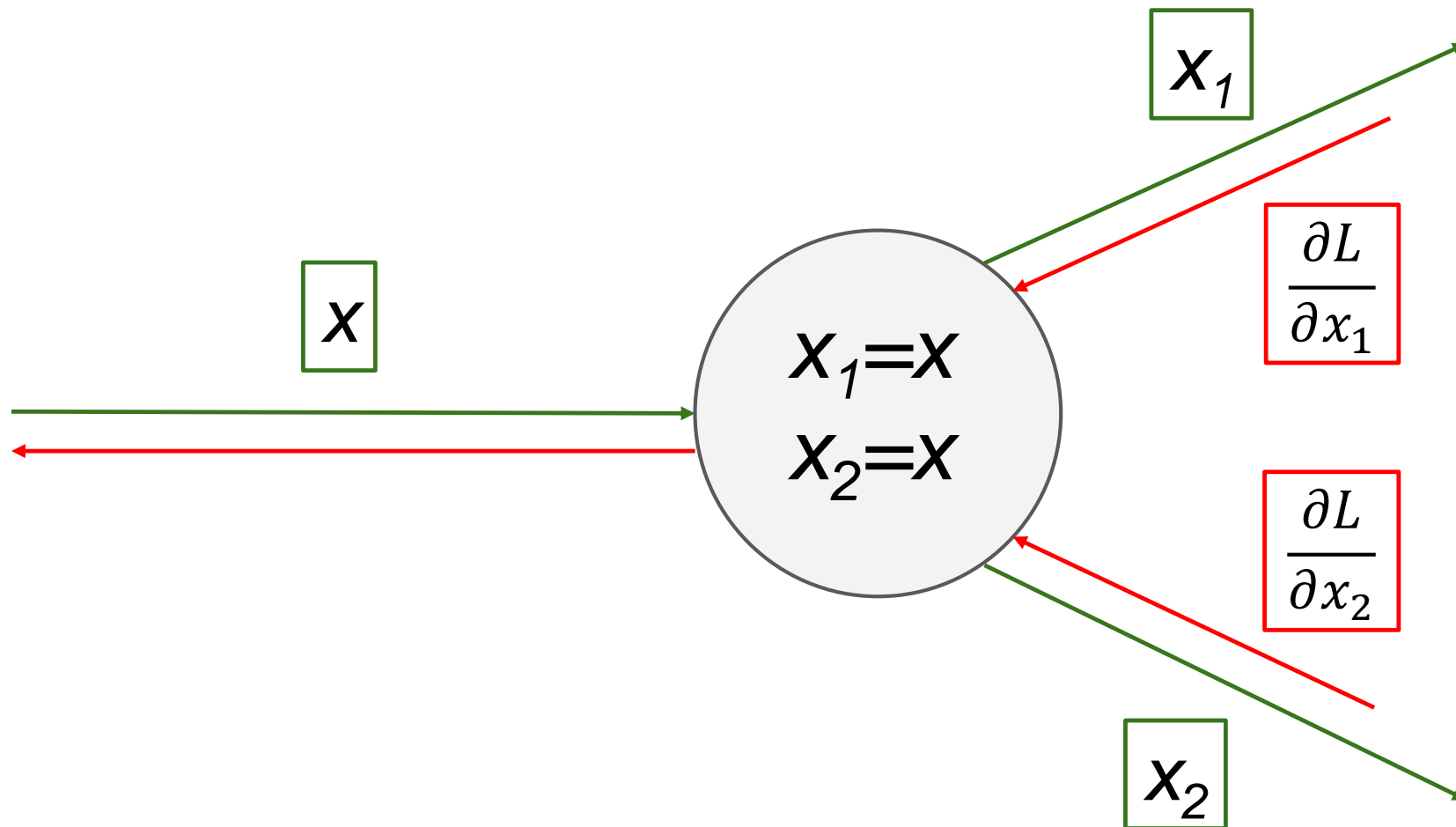
Обратный градиент при суммировании



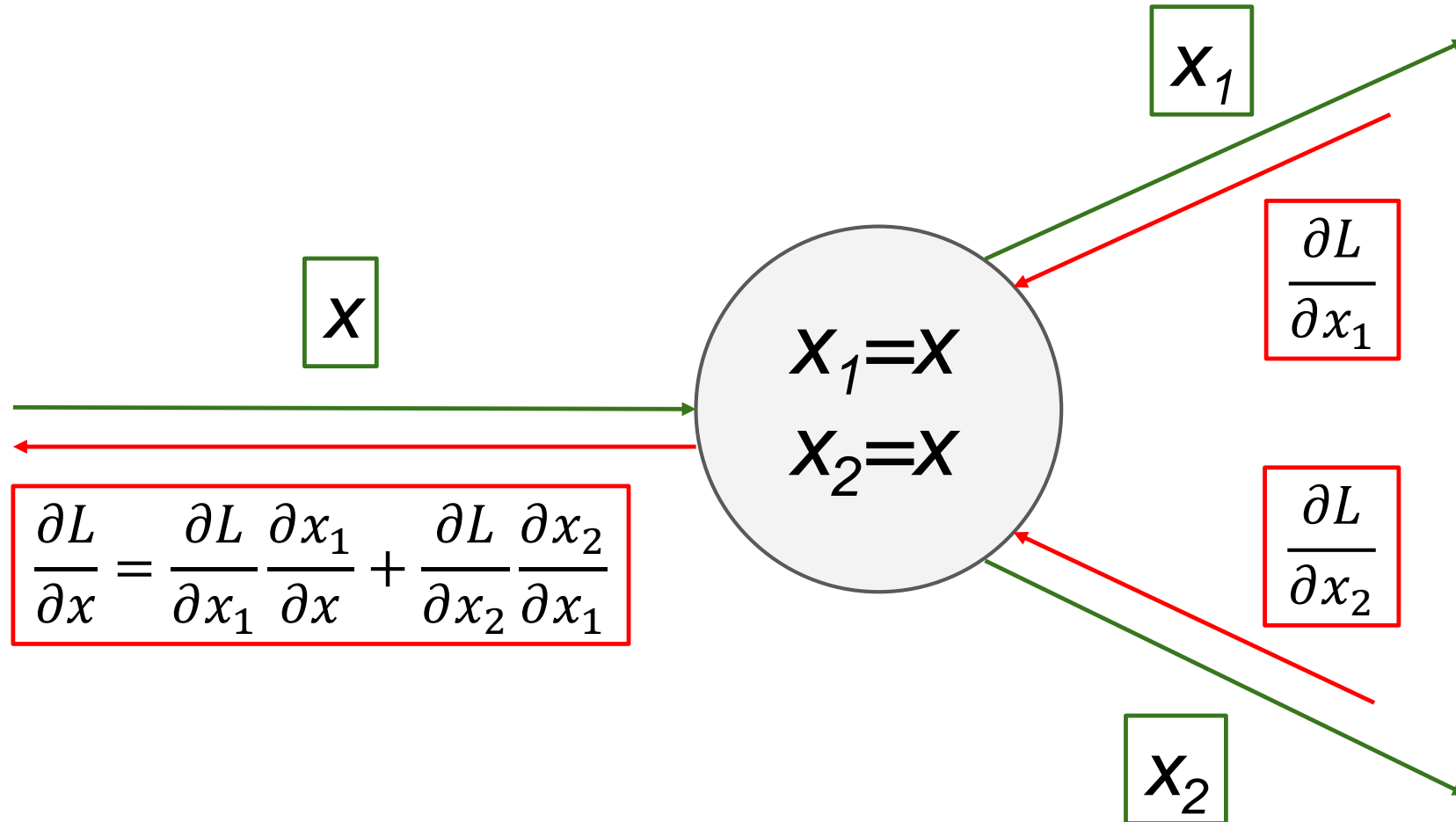
Обратный градиент при суммировании



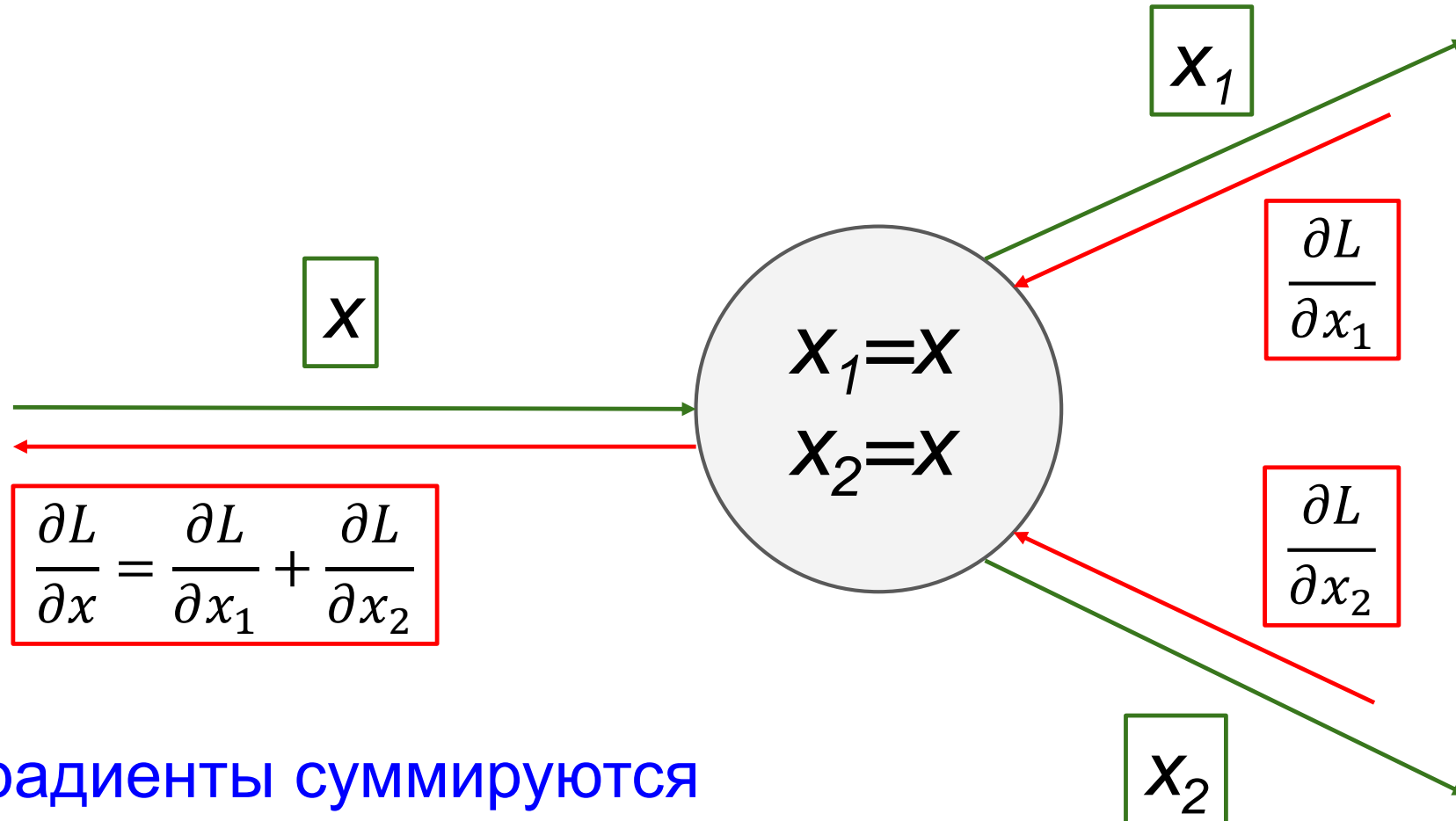
Обратный градиент при переиспользовании переменной



Обратный градиент при переиспользовании переменной

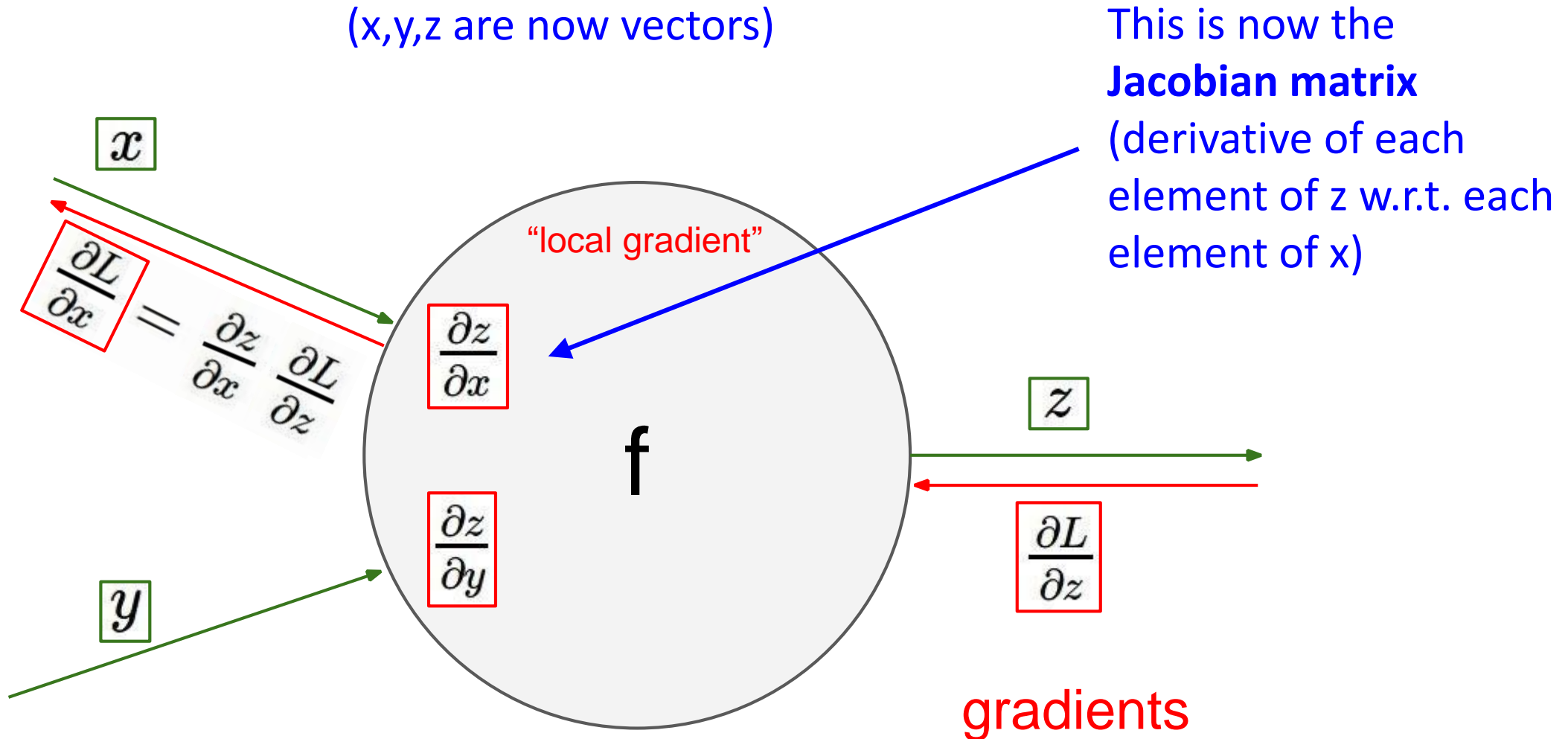


Обратный градиент при переиспользовании переменной

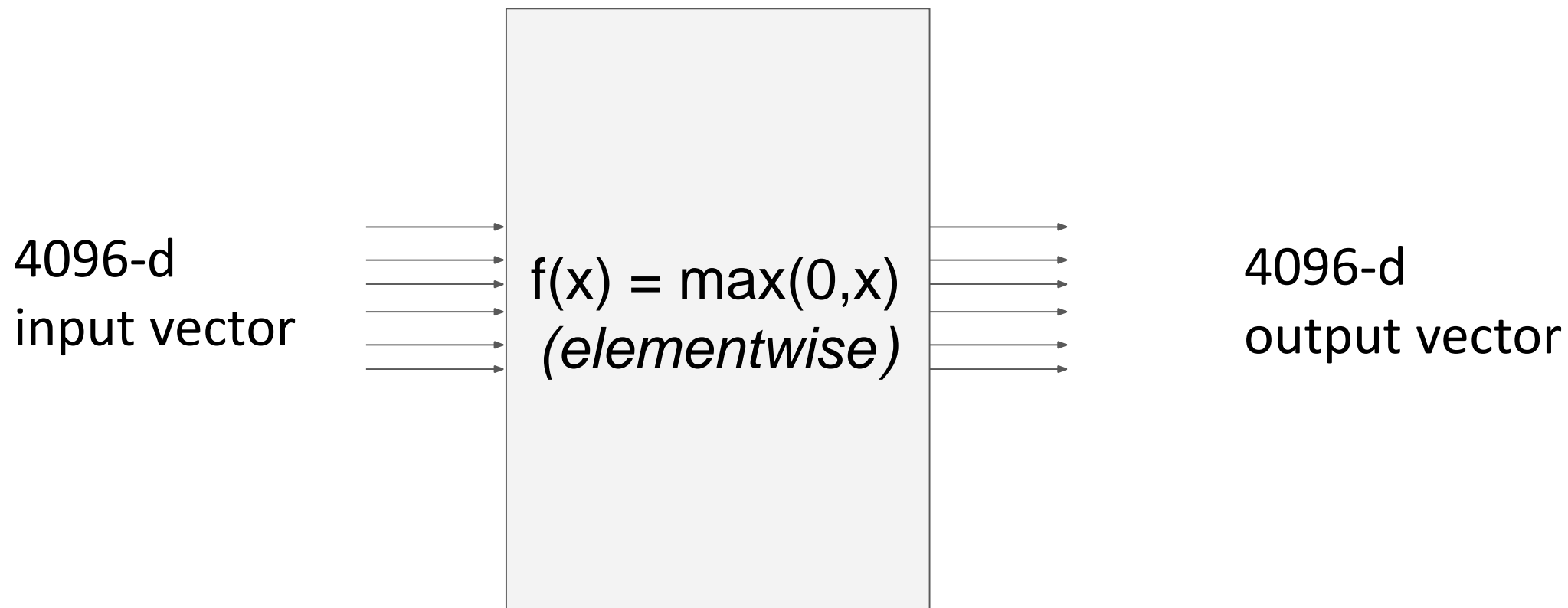


Backpropagation: векторный случай

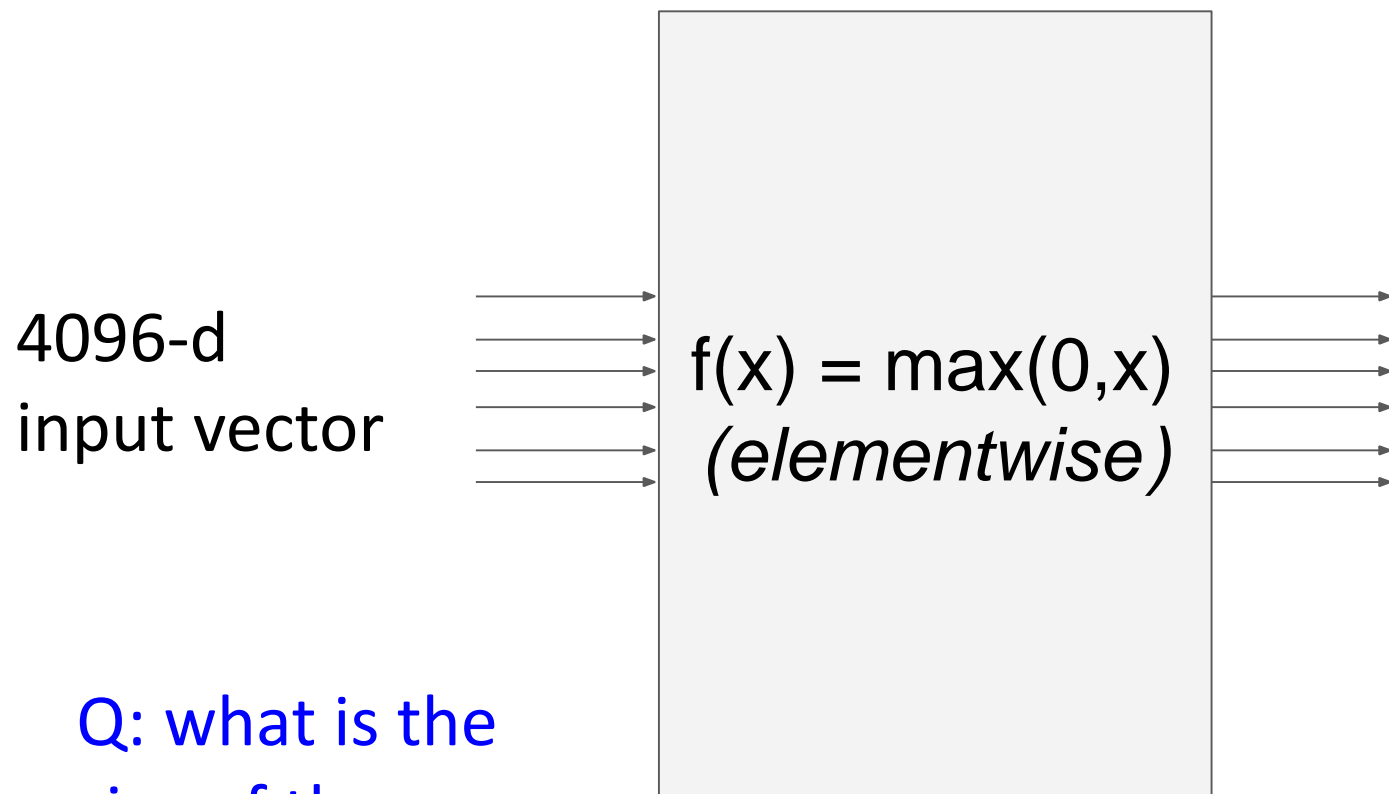
(x, y, z are now vectors)



Векторные операции



Векторные операции



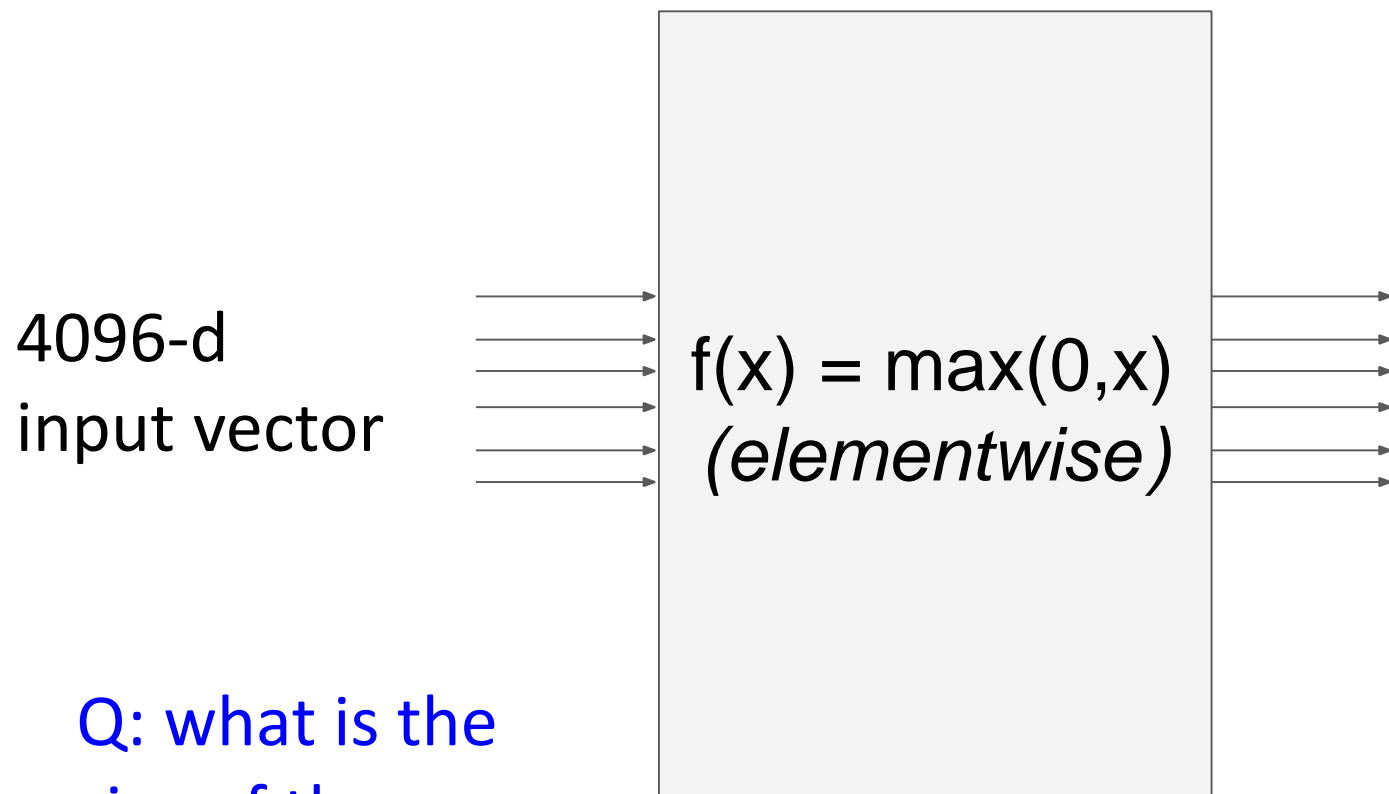
Q: what is the
size of the
Jacobian matrix?

$$\frac{\partial L}{\partial x} = \boxed{\frac{\partial f}{\partial x}} \frac{\partial L}{\partial f}$$

Jacobian matrix

4096-d
output vector

Векторные операции



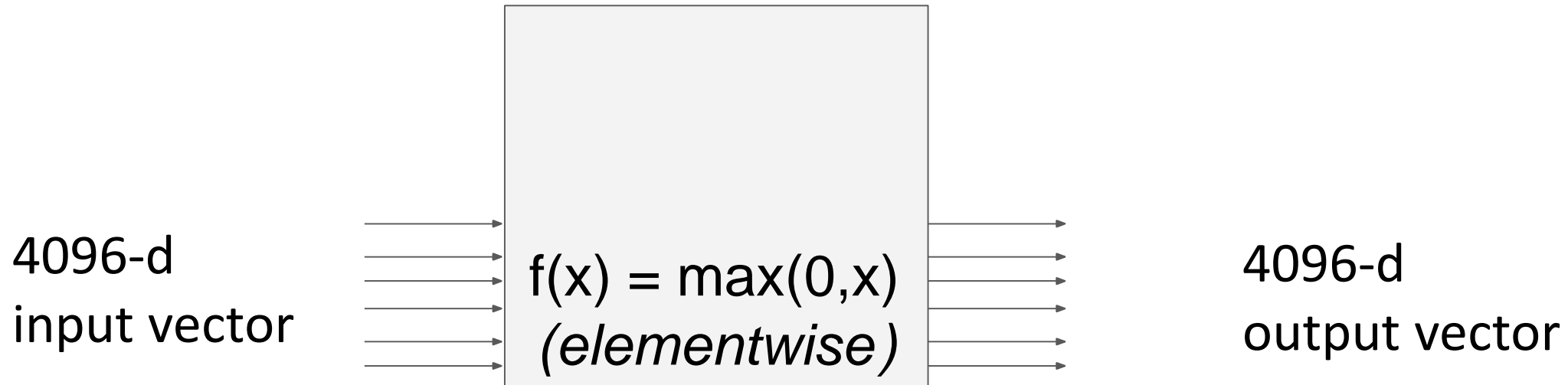
$$\frac{\partial L}{\partial x} = \boxed{\frac{\partial f}{\partial x}} \frac{\partial L}{\partial f}$$

Jacobian matrix

4096-d
output vector

Q: what is the
size of the
Jacobian matrix?
[4096 x 4096!]

Векторные операции

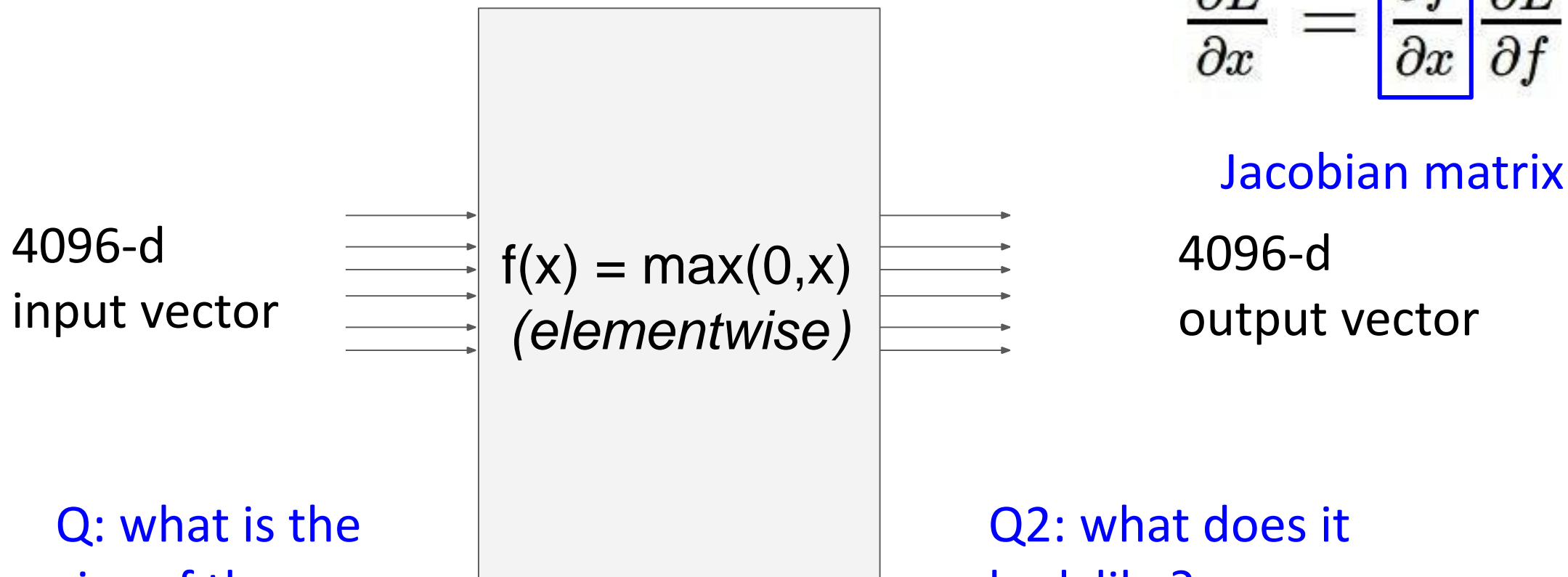


Q: what is the size of the Jacobian matrix?
[4096 x 4096!]

in practice we process an entire minibatch (e.g. 100) of examples at one time:

i.e. Jacobian would technically be a [409,600 x 409,600] matrix

Векторные операции

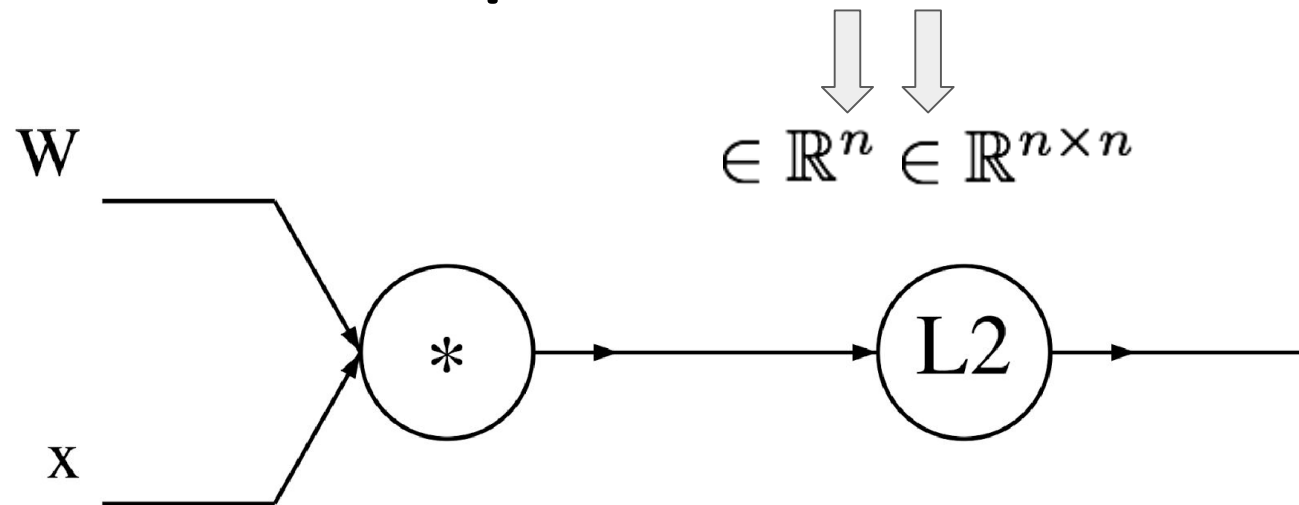


Q: what is the
size of the
Jacobian matrix?
[4096 x 4096!]

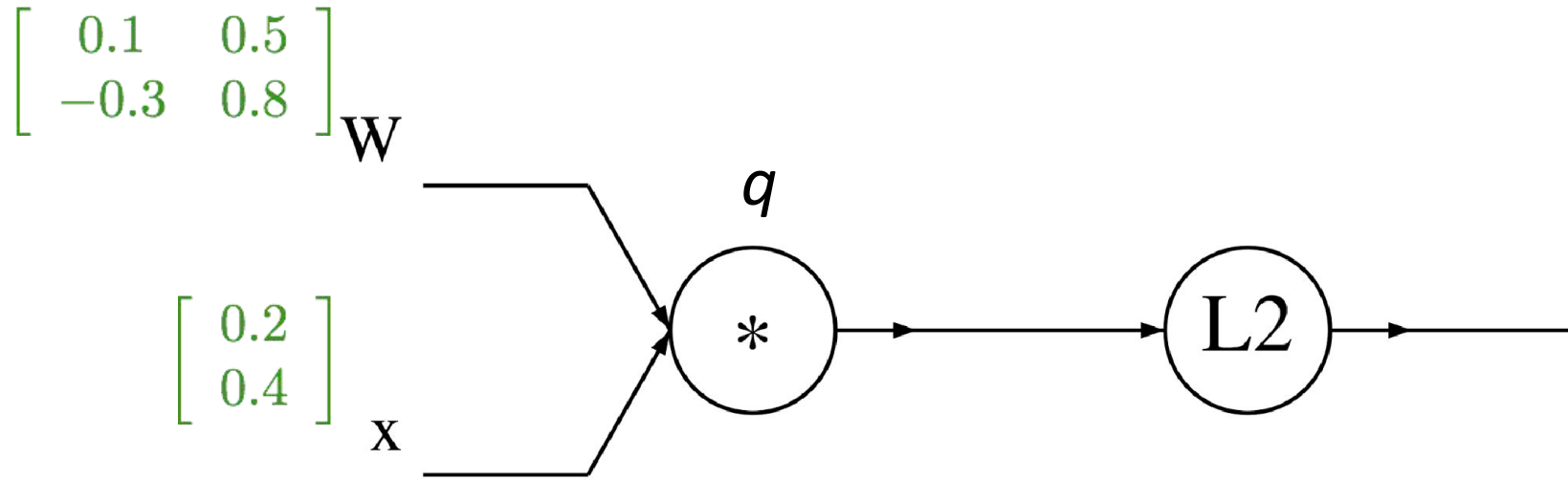
Q2: what does it
look like?

A vectorized example: $f(x, W) = ||W \cdot x||^2 = \sum_{i=1}^n (W \cdot x)_i^2$

A vectorized example: $f(x, W) = ||W \cdot x||^2 = \sum_{i=1}^n (W \cdot x)_i^2$



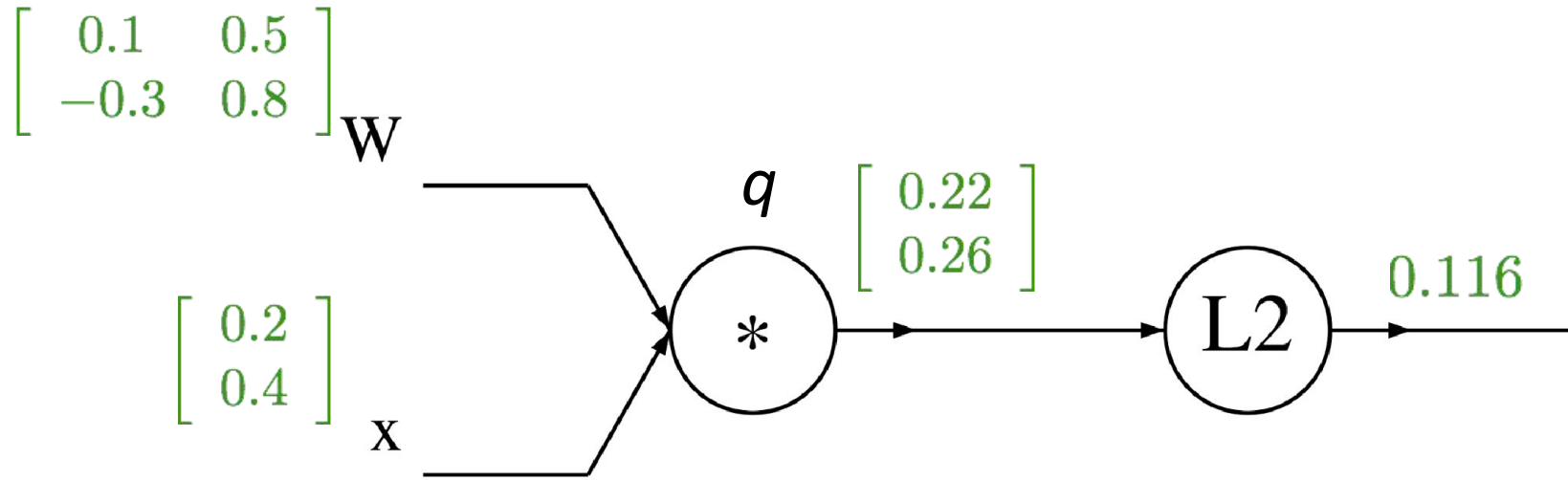
A vectorized example: $f(x, W) = ||W \cdot x||^2 = \sum_{i=1}^n (W \cdot x)_i^2$



$$q = W \cdot x = \begin{pmatrix} W_{1,1}x_1 + \cdots + W_{1,n}x_n \\ \vdots \\ W_{n,1}x_1 + \cdots + W_{n,n}x_n \end{pmatrix}$$

$$f(q) = ||q||^2 = q_1^2 + \cdots + q_n^2$$

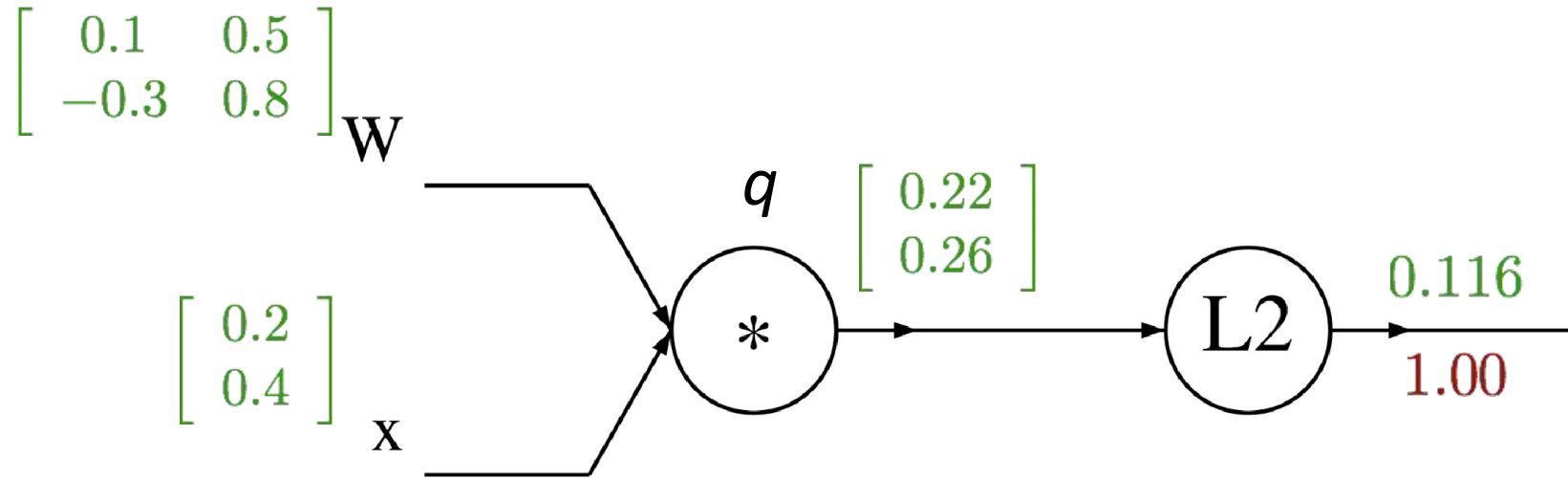
A vectorized example: $f(x, W) = ||W \cdot x||^2 = \sum_{i=1}^n (W \cdot x)_i^2$



$$q = W \cdot x = \begin{pmatrix} W_{1,1}x_1 + \cdots + W_{1,n}x_n \\ \vdots \\ W_{n,1}x_1 + \cdots + W_{n,n}x_n \end{pmatrix}$$

$$f(q) = ||q||^2 = q_1^2 + \cdots + q_n^2$$

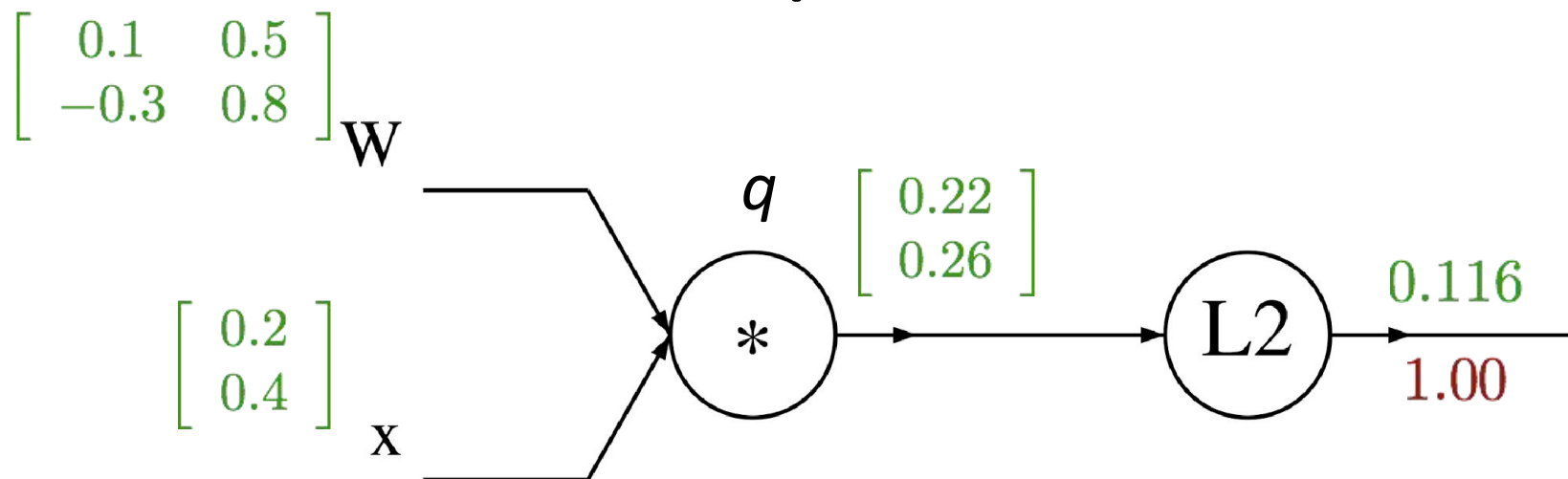
A vectorized example: $f(x, W) = ||W \cdot x||^2 = \sum_{i=1}^n (W \cdot x)_i^2$



$$q = W \cdot x = \begin{pmatrix} W_{1,1}x_1 + \cdots + W_{1,n}x_n \\ \vdots \\ W_{n,1}x_1 + \cdots + W_{n,n}x_n \end{pmatrix}$$

$$f(q) = ||q||^2 = q_1^2 + \cdots + q_n^2$$

A vectorized example: $f(x, W) = ||W \cdot x||^2 = \sum_{i=1}^n (W \cdot x)_i^2$



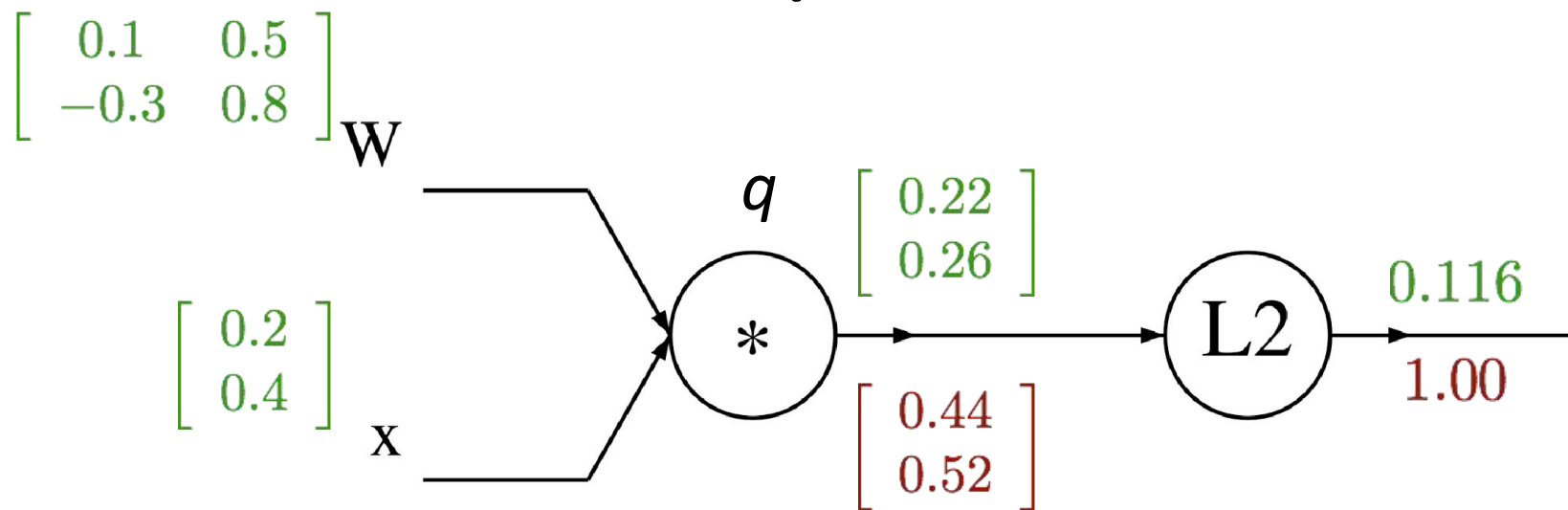
$$q = W \cdot x = \begin{pmatrix} W_{1,1}x_1 + \cdots + W_{1,n}x_n \\ \vdots \\ W_{n,1}x_1 + \cdots + W_{n,n}x_n \end{pmatrix}$$

$$\frac{\partial f}{\partial q_i} = 2q_i$$

$$\nabla_q f = 2q$$

$$f(q) = ||q||^2 = q_1^2 + \cdots + q_n^2$$

A vectorized example: $f(x, W) = ||W \cdot x||^2 = \sum_{i=1}^n (W \cdot x)_i^2$



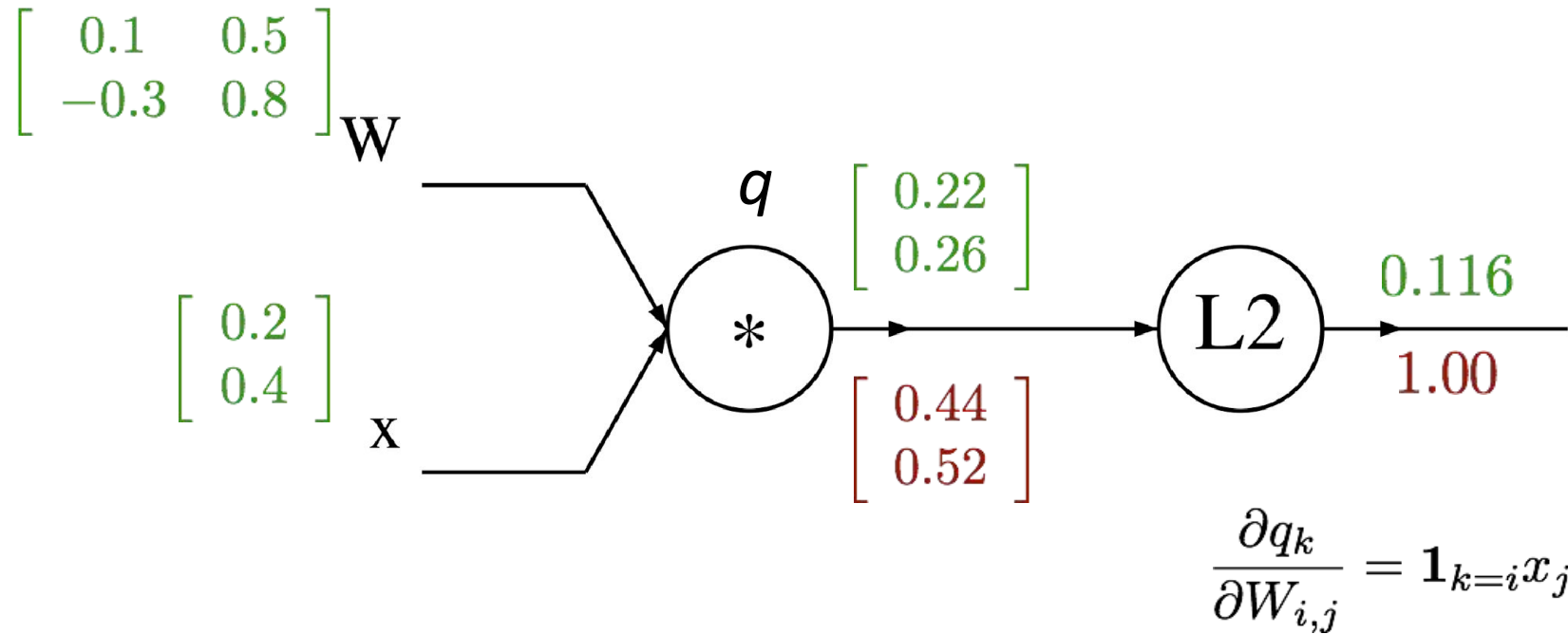
$$q = W \cdot x = \begin{pmatrix} W_{1,1}x_1 + \cdots + W_{1,n}x_n \\ \vdots \\ W_{n,1}x_1 + \cdots + W_{n,n}x_n \end{pmatrix}$$

$$\frac{\partial f}{\partial q_i} = 2q_i$$

$$\nabla_q f = 2q$$

$$f(q) = ||q||^2 = q_1^2 + \cdots + q_n^2$$

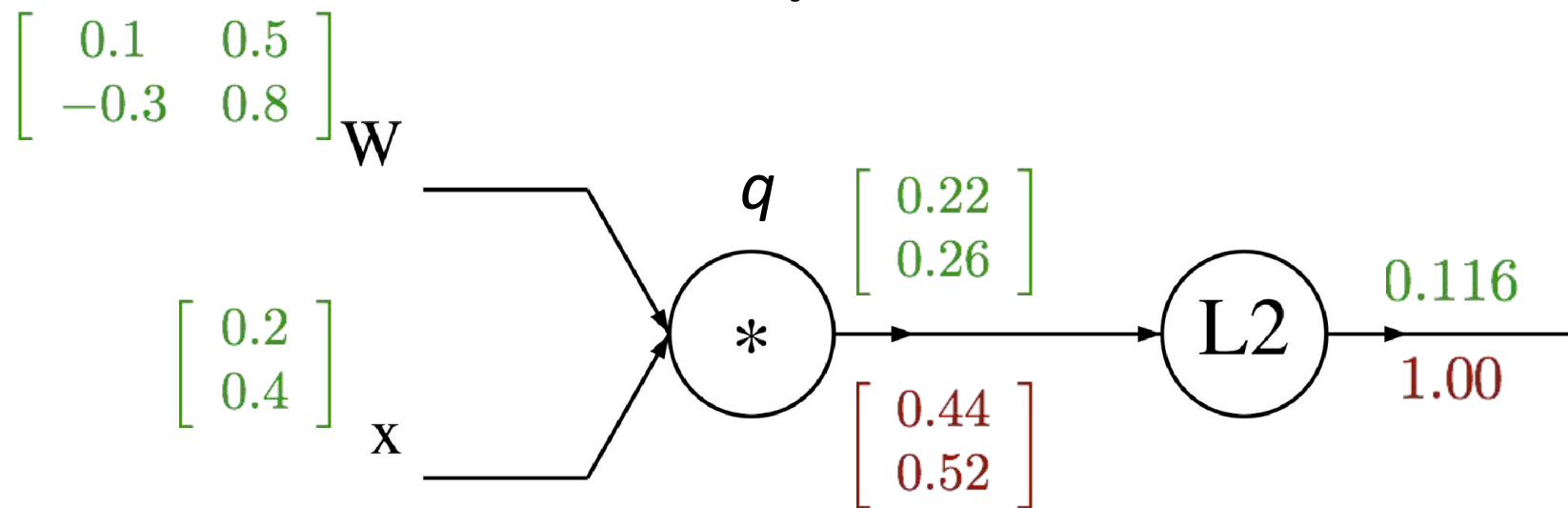
A vectorized example: $f(x, W) = ||W \cdot x||^2 = \sum_{i=1}^n (W \cdot x)_i^2$



$$q = W \cdot x = \begin{pmatrix} W_{1,1}x_1 + \cdots + W_{1,n}x_n \\ \vdots \\ W_{n,1}x_1 + \cdots + W_{n,n}x_n \end{pmatrix}$$

$$f(q) = ||q||^2 = q_1^2 + \cdots + q_n^2$$

A vectorized example: $f(x, W) = ||W \cdot x||^2 = \sum_{i=1}^n (W \cdot x)_i^2$



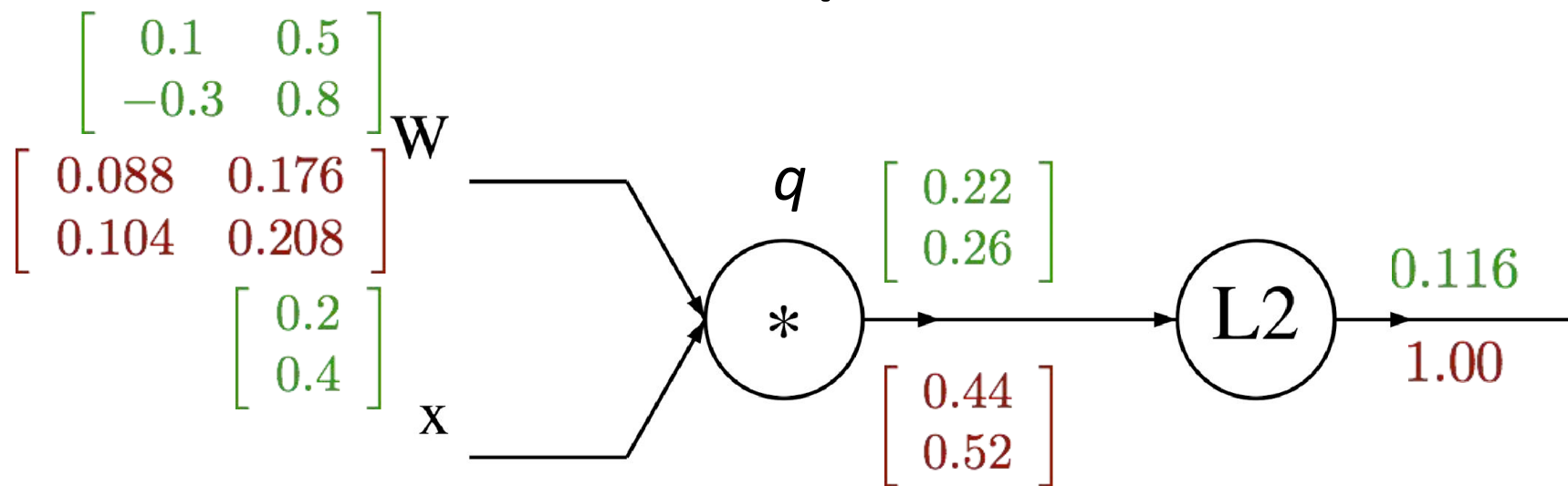
$$q = W \cdot x = \begin{pmatrix} W_{1,1}x_1 + \cdots + W_{1,n}x_n \\ \vdots \\ W_{n,1}x_1 + \cdots + W_{n,n}x_n \end{pmatrix}$$

$$f(q) = ||q||^2 = q_1^2 + \cdots + q_n^2$$

$$\frac{\partial q_k}{\partial W_{i,j}} = \mathbf{1}_{k=i} x_j$$

$$\begin{aligned} \frac{\partial f}{\partial W_{i,j}} &= \sum_k \frac{\partial f}{\partial q_k} \frac{\partial q_k}{\partial W_{i,j}} \\ &= \sum_k (2q_k) (\mathbf{1}_{k=i} x_j) \\ &= 2q_i x_j \end{aligned}$$

A vectorized example: $f(x, W) = ||W \cdot x||^2 = \sum_{i=1}^n (W \cdot x)_i^2$



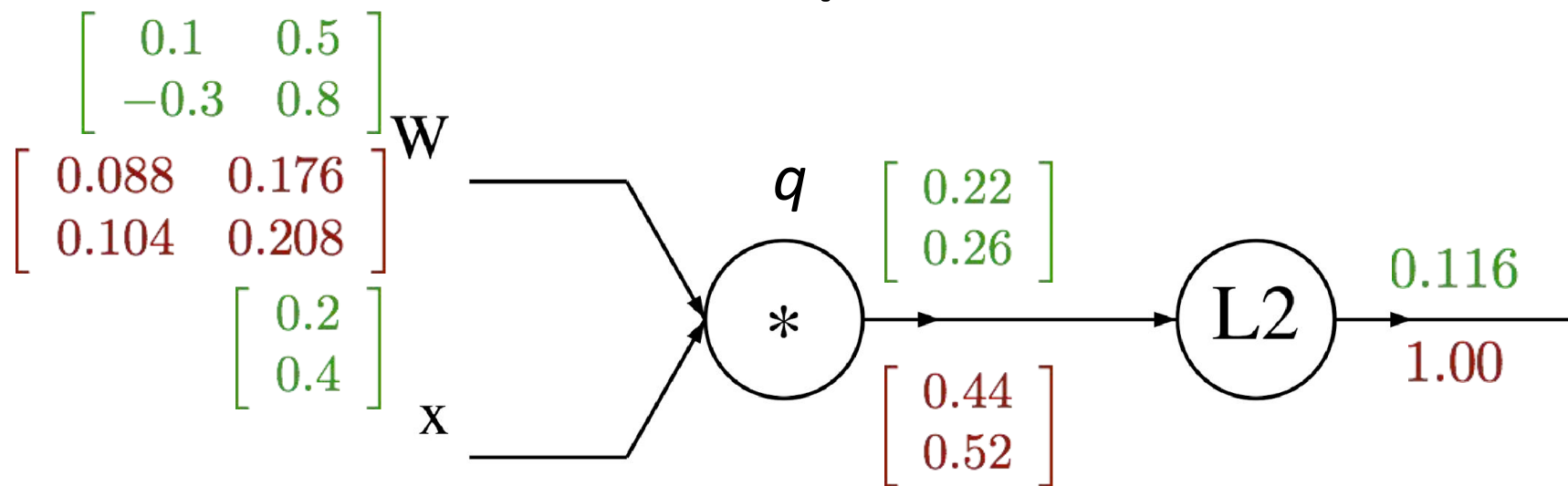
$$q = W \cdot x = \begin{pmatrix} W_{1,1}x_1 + \cdots + W_{1,n}x_n \\ \vdots \\ W_{n,1}x_1 + \cdots + W_{n,n}x_n \end{pmatrix}$$

$$f(q) = ||q||^2 = q_1^2 + \cdots + q_n^2$$

$$\frac{\partial q_k}{\partial W_{i,j}} = \mathbf{1}_{k=i} x_j$$

$$\begin{aligned} \frac{\partial f}{\partial W_{i,j}} &= \sum_k \frac{\partial f}{\partial q_k} \frac{\partial q_k}{\partial W_{i,j}} \\ &= \sum_k (2q_k) (\mathbf{1}_{k=i} x_j) \\ &= 2q_i x_j \end{aligned}$$

A vectorized example: $f(x, W) = ||W \cdot x||^2 = \sum_{i=1}^n (W \cdot x)_i^2$



$$\nabla_W f = 2q \cdot x^T$$

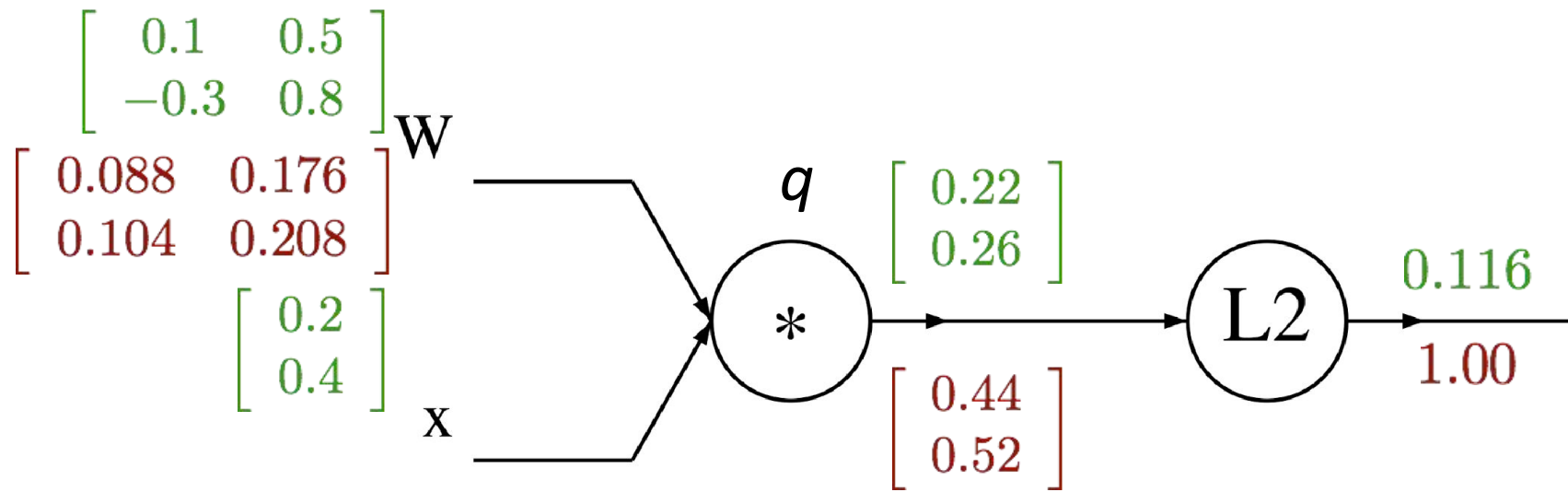
$$q = W \cdot x = \begin{pmatrix} W_{1,1}x_1 + \cdots + W_{1,n}x_n \\ \vdots \\ W_{n,1}x_1 + \cdots + W_{n,n}x_n \end{pmatrix}$$

$$\frac{\partial q_k}{\partial W_{i,j}} = \mathbf{1}_{k=i} x_j$$

$$\begin{aligned} \frac{\partial f}{\partial W_{i,j}} &= \sum_k \frac{\partial f}{\partial q_k} \frac{\partial q_k}{\partial W_{i,j}} \\ &= \sum_k (2q_k) (\mathbf{1}_{k=i} x_j) \\ &= 2q_i x_j \end{aligned}$$

$$f(q) = ||q||^2 = q_1^2 + \cdots + q_n^2$$

A vectorized example: $f(x, W) = ||W \cdot x||^2 = \sum_{i=1}^n (W \cdot x)_i^2$



$$\nabla_W f = 2q \cdot x^T$$

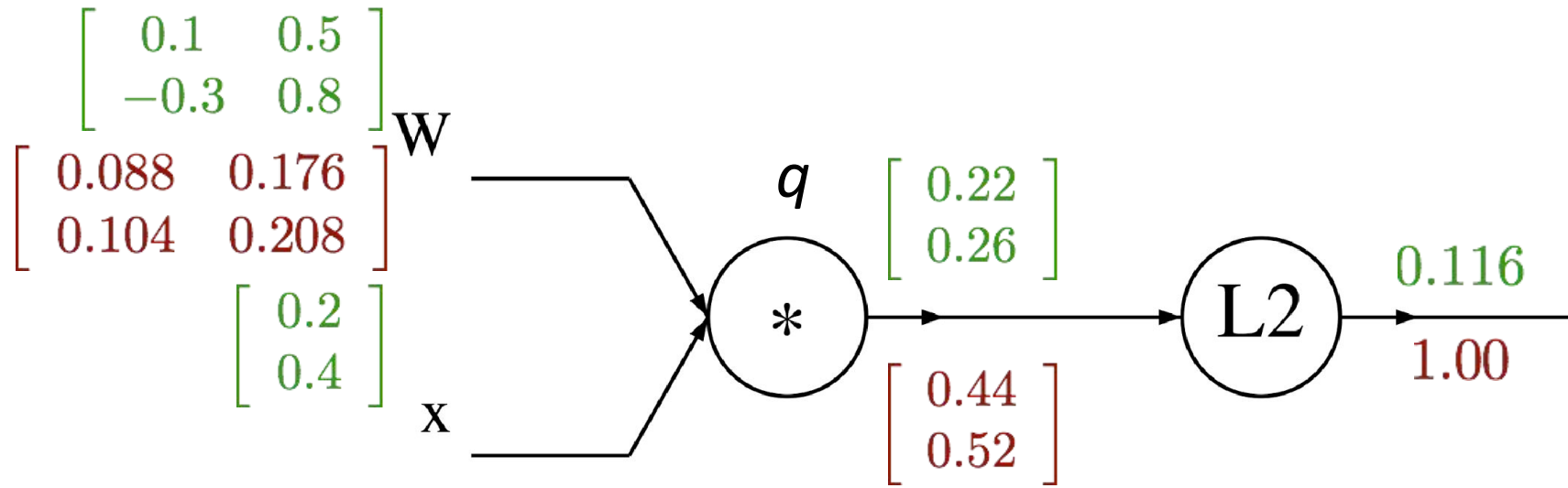
Всегда проверяйте
размерность
градиента

$$q = W \cdot x = \begin{pmatrix} W_{1,1}x_1 + \cdots + W_{1,n}x_n \\ \vdots \\ W_{n,1}x_1 + \cdots + W_{n,n}x_n \end{pmatrix}$$

$$f(q) = ||q||^2 = q_1^2 + \cdots + q_n^2$$

Размерность
градиента должна
быть такой-же как
размерность
переменной, для
которой он
вычисляется

A vectorized example: $f(x, W) = ||W \cdot x||^2 = \sum_{i=1}^n (W \cdot x)_i^2$

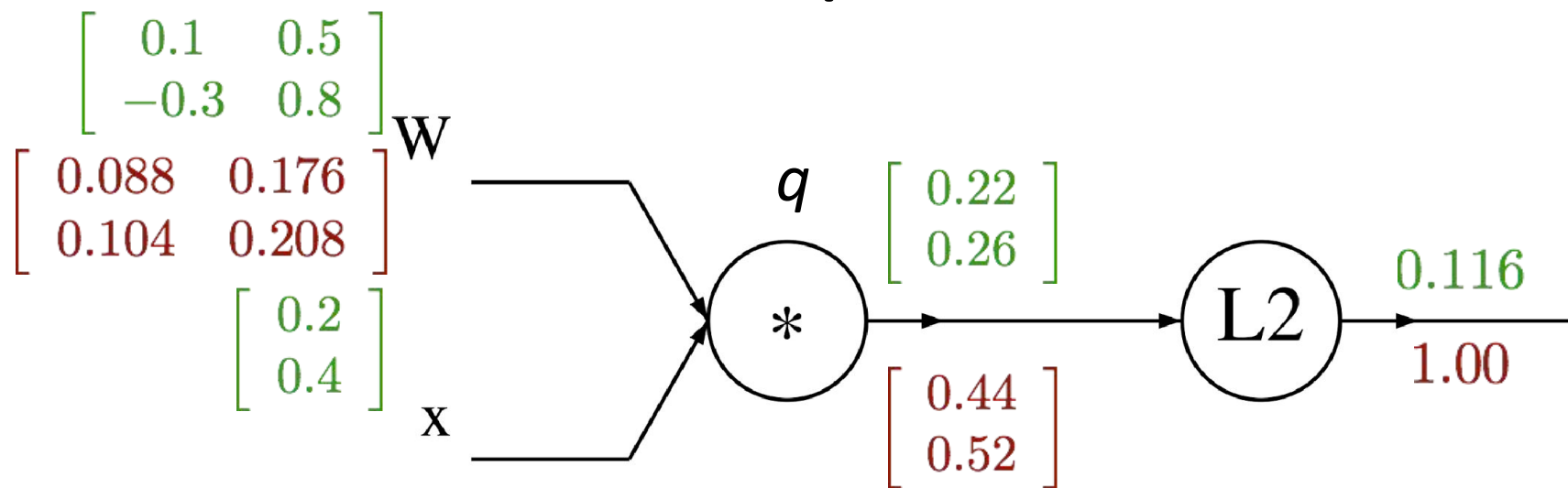


$$q = W \cdot x = \begin{pmatrix} W_{1,1}x_1 + \cdots + W_{1,n}x_n \\ \vdots \\ W_{n,1}x_1 + \cdots + W_{n,n}x_n \end{pmatrix}$$

$$\frac{\partial q_k}{\partial x_i} = W_{k,i}$$

$$f(q) = ||q||^2 = q_1^2 + \cdots + q_n^2$$

A vectorized example: $f(x, W) = ||W \cdot x||^2 = \sum_{i=1}^n (W \cdot x)_i^2$



$$q = W \cdot x = \begin{pmatrix} W_{1,1}x_1 + \cdots + W_{1,n}x_n \\ \vdots \\ W_{n,1}x_1 + \cdots + W_{n,n}x_n \end{pmatrix}$$

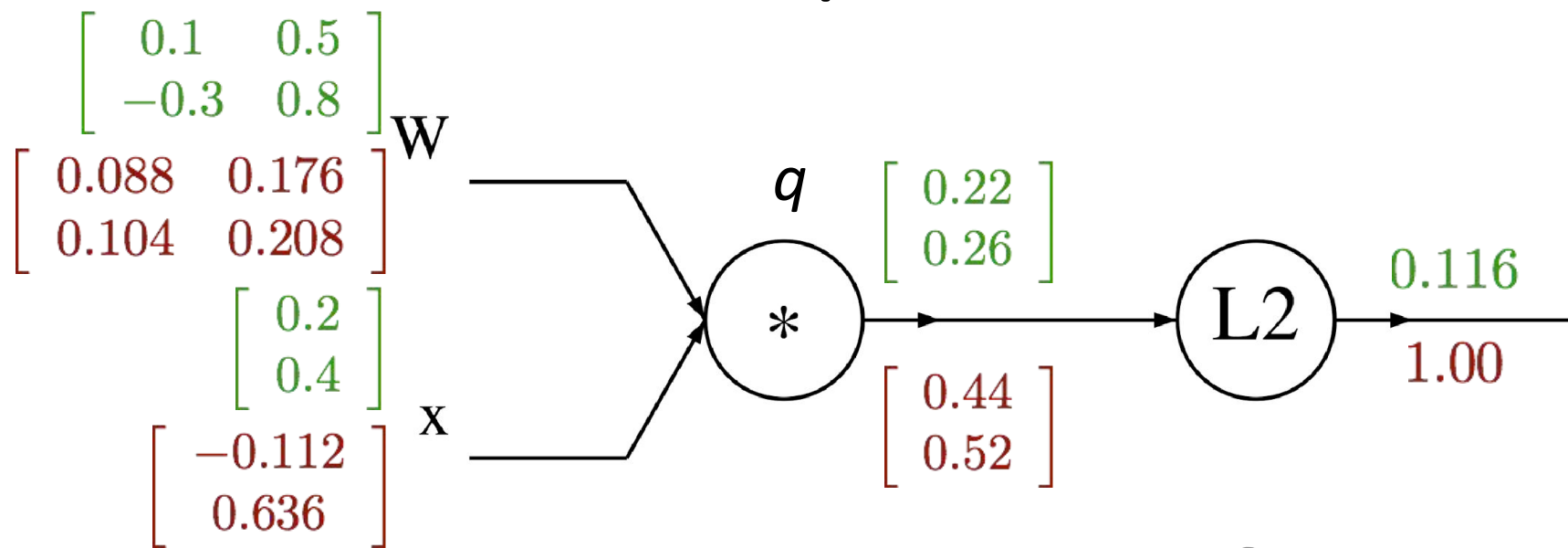
$$f(q) = ||q||^2 = q_1^2 + \cdots + q_n^2$$

$$\frac{\partial q_k}{\partial x_i} = W_{k,i}$$

$$\frac{\partial f}{\partial x_i} = \sum_k \frac{\partial f}{\partial q_k} \frac{\partial q_k}{\partial x_i}$$

$$\frac{\partial f}{\partial x_i} = \sum_k 2q_k W_{k,i}$$

A vectorized example: $f(x, W) = ||W \cdot x||^2 = \sum_{i=1}^n (W \cdot x)_i^2$



$$\nabla_x f = 2W^T \cdot q$$

$$q = W \cdot x = \begin{pmatrix} W_{1,1}x_1 + \cdots + W_{1,n}x_n \\ \vdots \\ W_{n,1}x_1 + \cdots + W_{n,n}x_n \end{pmatrix}$$

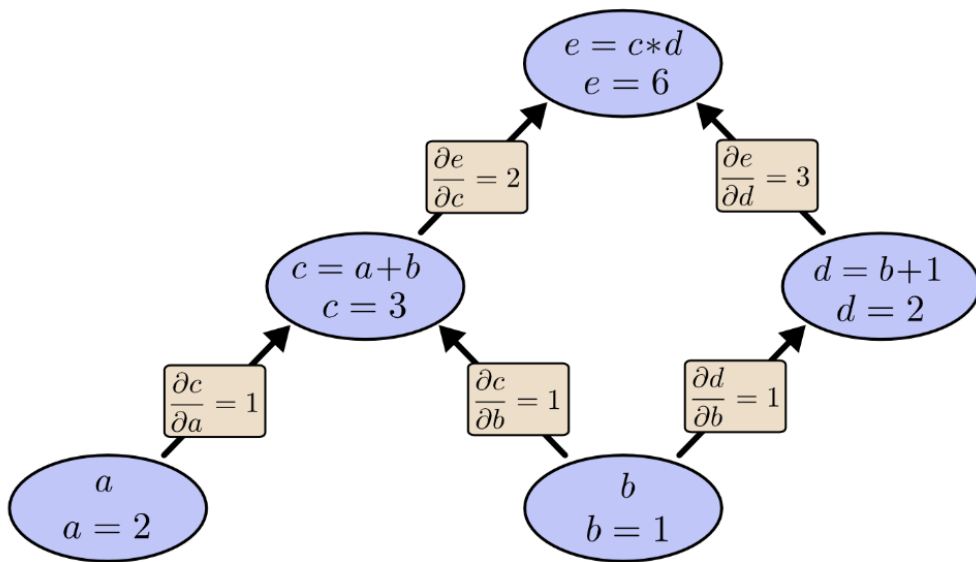
$$f(q) = ||q||^2 = q_1^2 + \cdots + q_n^2$$

$$\frac{\partial q_k}{\partial x_i} = W_{k,i}$$

$$\frac{\partial f}{\partial x_i} = \sum_k \frac{\partial f}{\partial q_k} \frac{\partial q_k}{\partial x_i}$$

$$\frac{\partial f}{\partial x_i} = \sum_k 2q_k W_{k,i}$$

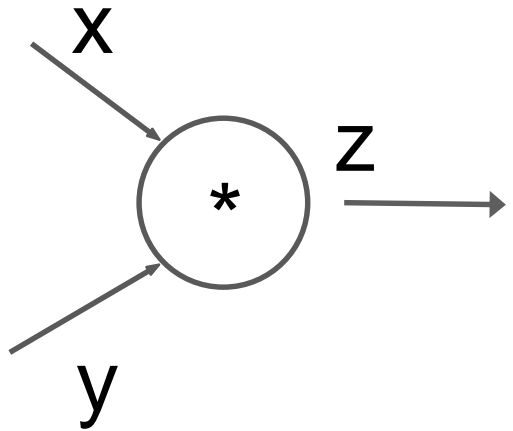
Modularized implementation: forward / backward API



Graph (or Net) object (*rough psuedo code*)

```
class ComputationalGraph(object):  
    #...  
    def forward(inputs):  
        # 1. [pass inputs to input gates...]  
        # 2. forward the computational graph:  
        for gate in self.graph.nodes_topologically_sorted():  
            gate.forward()  
        return loss # the final gate in the graph outputs the loss  
    def backward():  
        for gate in reversed(self.graph.nodes_topologically_sorted()):  
            gate.backward() # little piece of backprop (chain rule applied)  
        return inputs_gradients
```

Modularized implementation: forward / backward API



(x,y,z are scalars)

```
class MultiplyGate(object):  
    def forward(x,y):  
        z = x*y  
        return z  
    def backward(dz):  
        # dx = ... #todo  
        # dy = ... #todo  
        return [dx, dy]
```

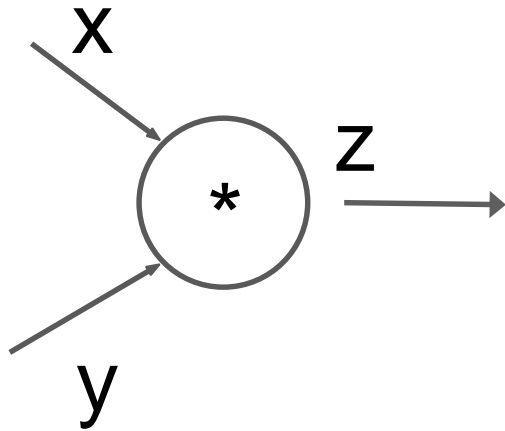
$$\frac{\partial L}{\partial z}$$

An arrow points from this box to the `dz` parameter in the `backward` method of the `MultiplyGate` class.

$$\frac{\partial L}{\partial x}$$

An arrow points from this box to the `dx` element in the `return [dx, dy]` statement of the `backward` method of the `MultiplyGate` class.

Modularized implementation: forward / backward API




(x,y,z are scalars)

```
class MultiplyGate(object):  
    def forward(x,y):  
        z = x*y  
        self.x = x # must keep these around!  
        self.y = y  
        return z  
    def backward(dz):  
        dx = self.y * dz # [dz/dx * dL/dz]  
        dy = self.x * dz # [dz/dy * dL/dz]  
        return [dx, dy]
```



























Example: Caffe layers

Branch: master **caffe** / src / caffe / layers /

Create new fileUpload filesFind fileHistory

 shelhamer committed on GitHub Merge pull request #4630 from B1Gene/load_hdf5_fix ... Latest commit e687a71 21 days ago

..

 absval_layer.cpp	dismantle layer headers	a year ago
 absval_layer.cu	dismantle layer headers	a year ago
 accuracy_layer.cpp	dismantle layer headers	a year ago
 argmax_layer.cpp	dismantle layer headers	a year ago
 base_conv_layer.cpp	enable dilated deconvolution	a year ago
 base_data_layer.cpp	Using default from proto for prefetch	3 months ago
 base_data_layer.cu	Switched multi-GPU to NCCL	3 months ago
 batch_norm_layer.cpp	Add missing spaces besides equal signs in batch_norm_layer.cpp	4 months ago
 batch_norm_layer.cu	dismantle layer headers	a year ago
 batch_reindex_layer.cpp	dismantle layer headers	a year ago
 batch_reindex_layer.cu	dismantle layer headers	a year ago
 bias_layer.cpp	Remove incorrect cast of gemm int arg to Dtype in BiasLayer	a year ago
 bias_layer.cu	Separation and generalization of ChannelwiseAffineLayer into BiasLayer	a year ago
 bnll_layer.cpp	dismantle layer headers	a year ago
 bnll_layer.cu	dismantle layer headers	a year ago
 concat_layer.cpp	dismantle layer headers	a year ago
 concat_layer.cu	dismantle layer headers	a year ago
 contrastive_loss_layer.cpp	dismantle layer headers	a year ago
 contrastive_loss_layer.cu	dismantle layer headers	a year ago
 conv_layer.cpp	add support for 2D dilated convolution	a year ago
 conv_layer.cu	dismantle layer headers	a year ago
 crop_layer.cpp	remove redundant operations in Crop layer (#5138)	2 months ago
 crop_layer.cu	remove redundant operations in Crop layer (#5138)	2 months ago
 cudnn_conv_layer.cpp	dismantle layer headers	a year ago
 cudnn_conv_layer.cu	Add cuDNN v5 support, drop cuDNN v3 support	11 months ago

=



Caffe Sigmoid Layer

```
1 #include <cmath>
2 #include <vector>
3
4 #include "caffe/layers/sigmoid_layer.hpp"
5
6 namespace caffe {
7
8 template <typename Dtype>
9 inline Dtype sigmoid(Dtype x) {
10     return 1. / (1. + exp(-x));
11 }
12
13 template <typename Dtype>
14 void SigmoidLayer<Dtype>::Forward_cpu(const vector<Blob<Dtype>*>& bottom,
15     const vector<Blob<Dtype>*>& top) {
16     const Dtype* bottom_data = bottom[0]->cpu_data();
17     Dtype* top_data = top[0]->mutable_cpu_data();
18     const int count = bottom[0]->count();
19     for (int i = 0; i < count; ++i) {
20         top_data[i] = sigmoid(bottom_data[i]);
21     }
22 }
23
24 template <typename Dtype>
25 void SigmoidLayer<Dtype>::Backward_cpu(const vector<Blob<Dtype>*>& top,
26     const vector<bool>& propagate_down,
27     const vector<Blob<Dtype>*>& bottom) {
28     if (propagate_down[0]) {
29         const Dtype* top_data = top[0]->cpu_data();
30         const Dtype* top_diff = top[0]->cpu_diff();
31         Dtype* bottom_diff = bottom[0]->mutable_cpu_diff();
32         const int count = bottom[0]->count();
33         for (int i = 0; i < count; ++i) {
34             const Dtype sigmoid_x = top_data[i];
35             bottom_diff[i] = top_diff[i] * sigmoid_x * (1. - sigmoid_x);
36         }
37     }
38 }
39
40 #ifdef CPU_ONLY
41 STUB_GPU(SigmoidLayer);
42 #endif
43
44 INSTANTIATE_CLASS(SigmoidLayer);
45
46 } // namespace caffe
```

forward()

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

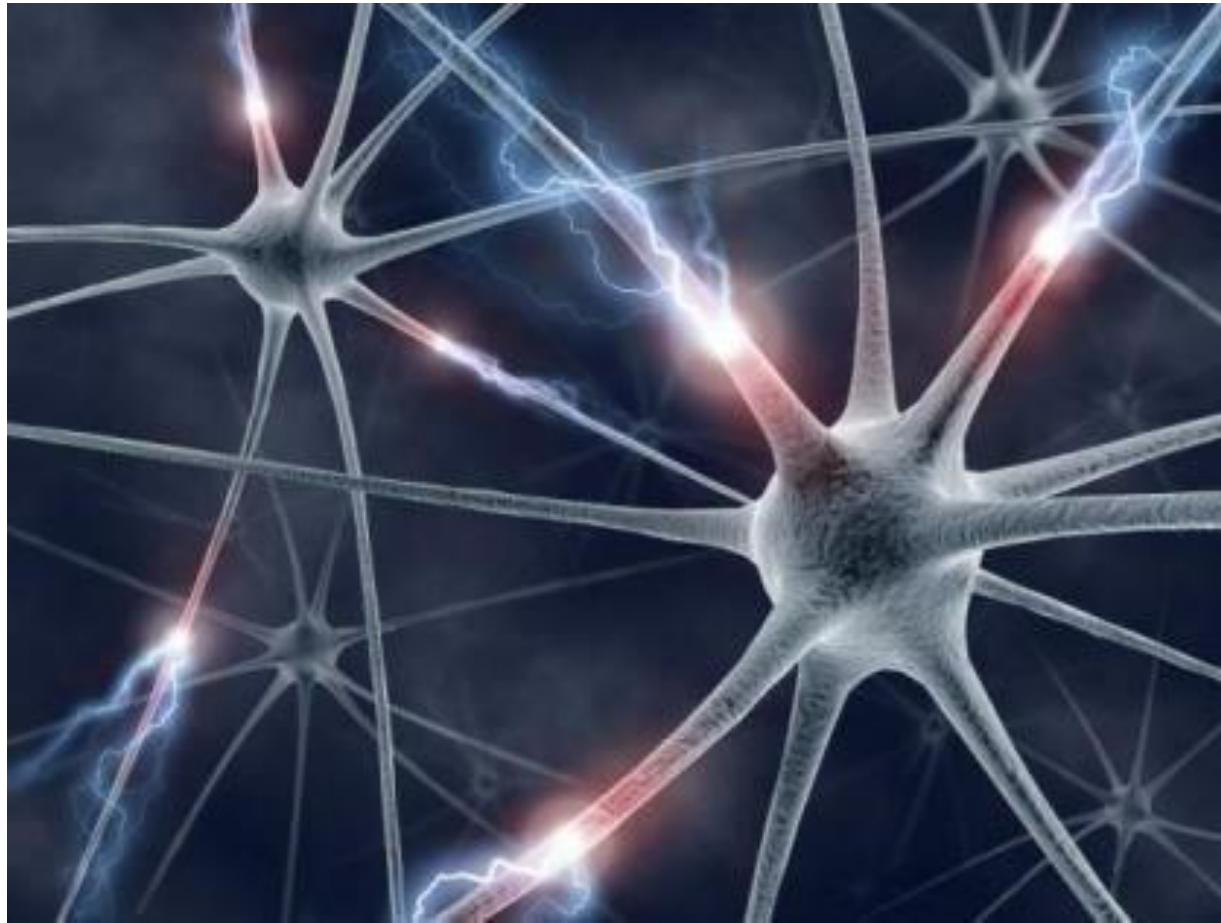
backward()

$$(1 - \sigma(x)) \sigma(x) * \text{top_diff} \text{ (chain rule)}$$

Backpropagation summary

- neural nets will be very large: impractical to write down gradient formula by hand for all parameters
- **backpropagation** = recursive application of the chain rule along a computational graph to compute the gradients of all inputs/parameters/intermediates
- implementations maintain a graph structure, where the nodes implement the **forward()** / **backward()** API
- **forward**: compute result of an operation and save any intermediates needed for gradient computation in memory
- **backward**: apply the chain rule to compute the gradient of the loss function with respect to the inputs

Neural Networks



Neural Networks

(**Before**) Linear score function: $f = Wx$

Neural Networks

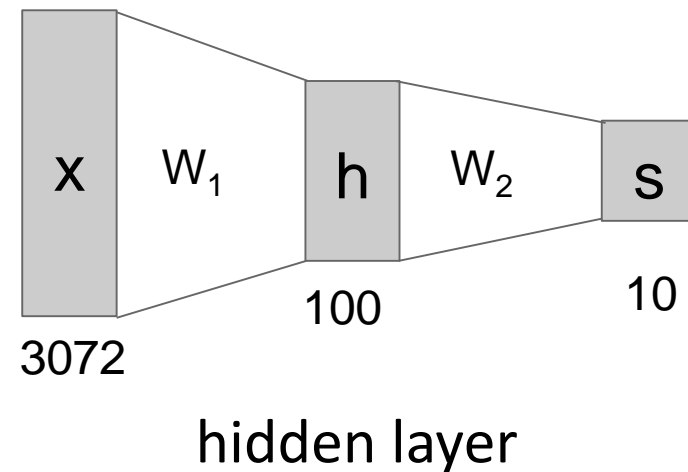
(**Before**) Linear score function: $f = Wx$

(**Now**) 2-layer Neural Network: $f = W_2 \max(0, W_1 x)$

Neural Networks

(**Before**) Linear score function: $f = Wx$

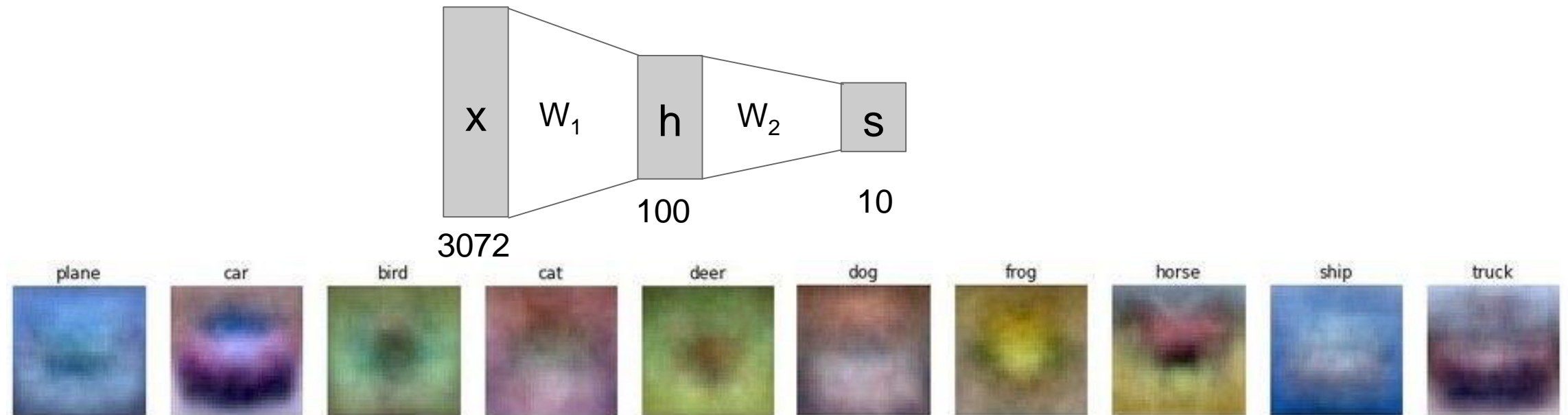
(**Now**) 2-layer Neural Network: $f = W_2 \max(0, W_1 x)$



Neural Networks

(**Before**) Linear score function: $f = Wx$

(**Now**) 2-layer Neural Network: $f = W_2 \max(0, W_1 x)$



Neural Networks

(**Before**) Linear score function: $f = Wx$

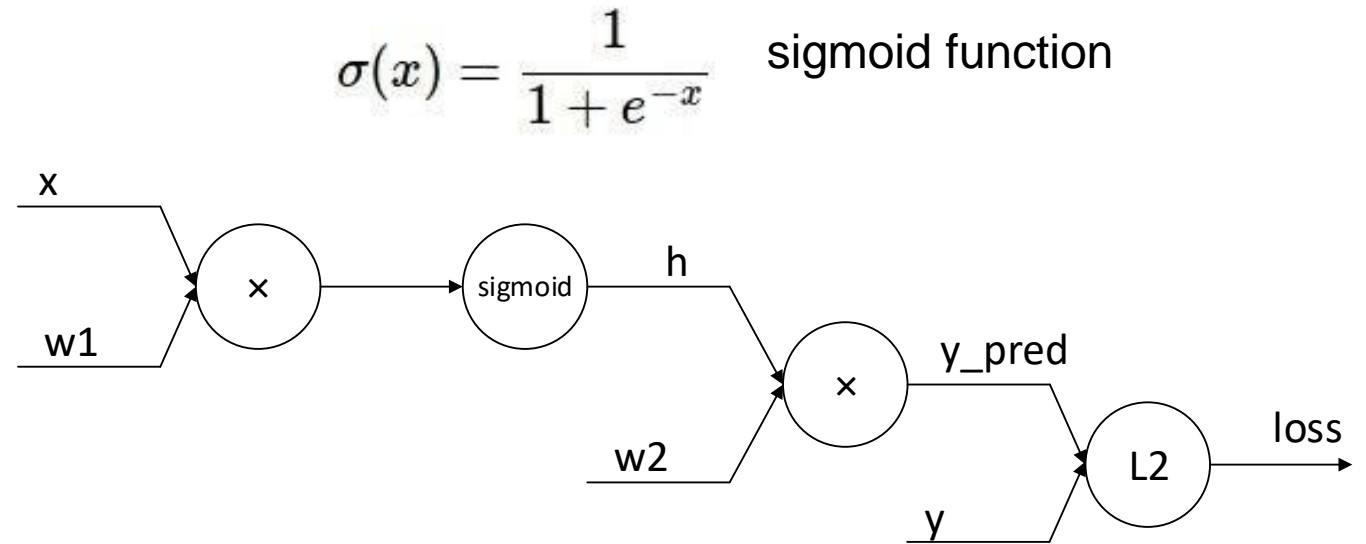
(**Now**) 2-layer Neural Network: $f = W_2 \max(0, W_1 x)$

we can go deeper

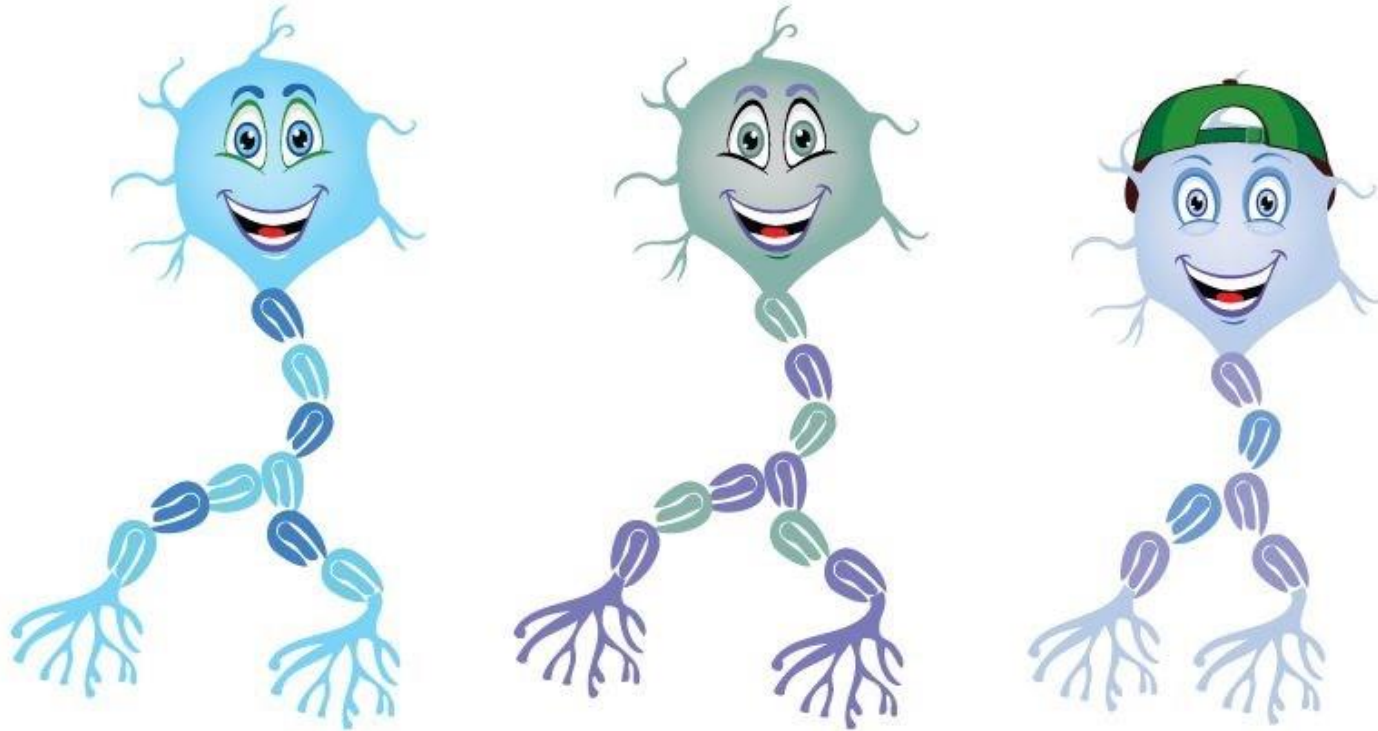
3-layer Neural Network $f = W_3 \max(0, W_2 \max(0, W_1 x))$

Full implementation of training a 2-layer Neural Network needs ~20 lines:

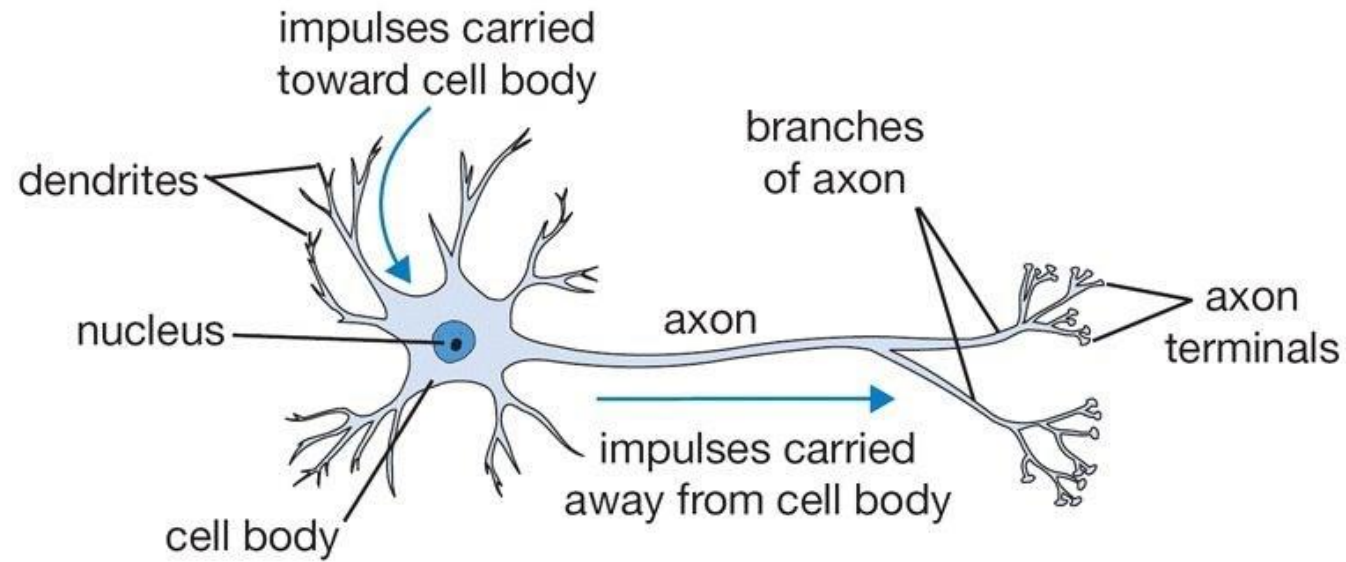
```
1 import numpy as np
2 from numpy.random import randn
3
4 N, D_in, H, D_out = 64, 1000, 100, 10
5 x, y = randn(N, D_in), randn(N, D_out)
6 w1, w2 = randn(D_in, H), randn(H, D_out)
7
8 for t in range(2000):
9     h = 1 / (1 + np.exp(-x.dot(w1)))
10    y_pred = h.dot(w2)
11    loss = np.square(y_pred - y).sum()
12    print(t, loss)
13
14    grad_y_pred = 2.0 * (y_pred - y)
15    grad_w2 = h.T.dot(grad_y_pred)
16    grad_h = grad_y_pred.dot(w2.T)
17    grad_w1 = x.T.dot(grad_h * h * (1 - h))
18
19    w1 -= 1e-4 * grad_w1
20    w2 -= 1e-4 * grad_w2
```



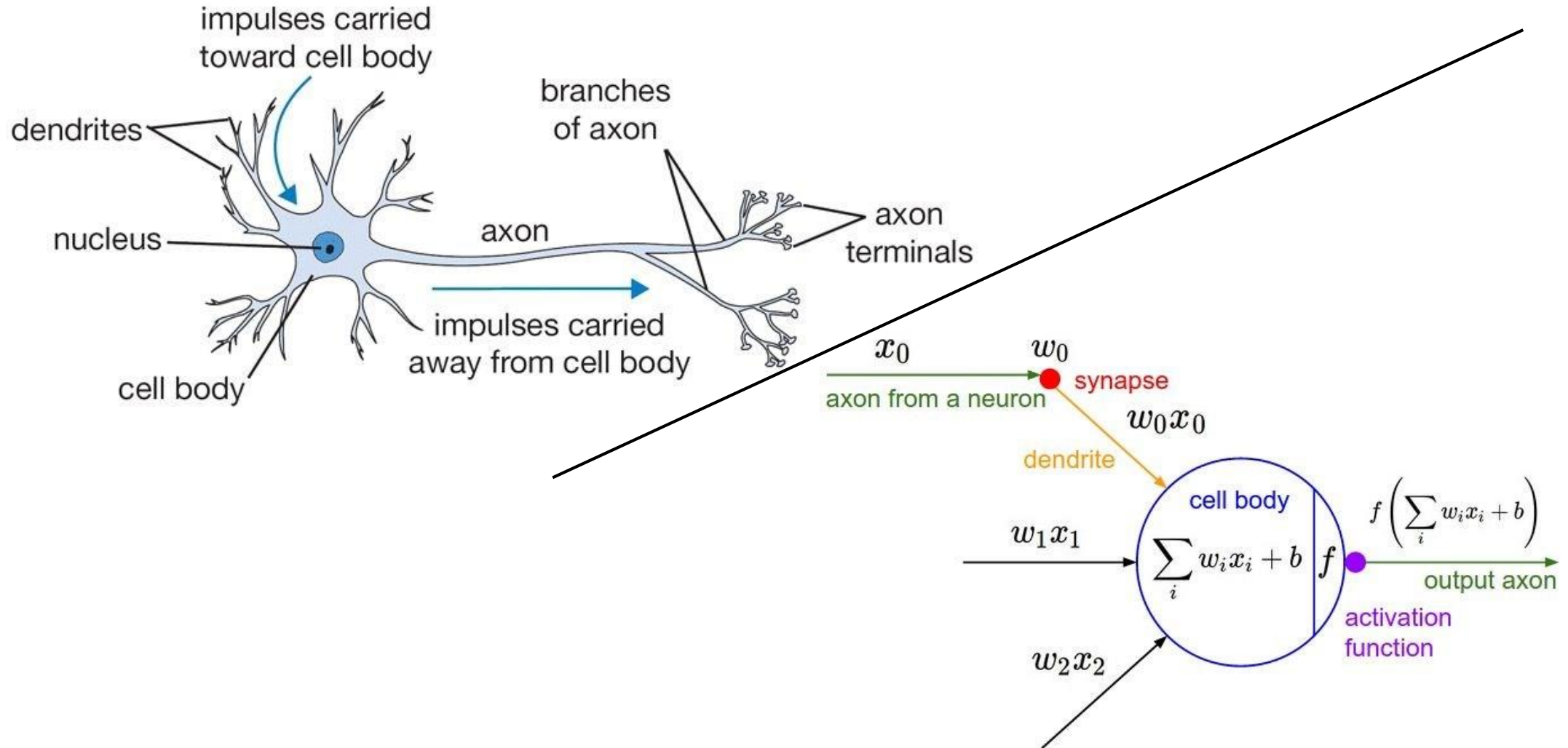
Artificial neuron



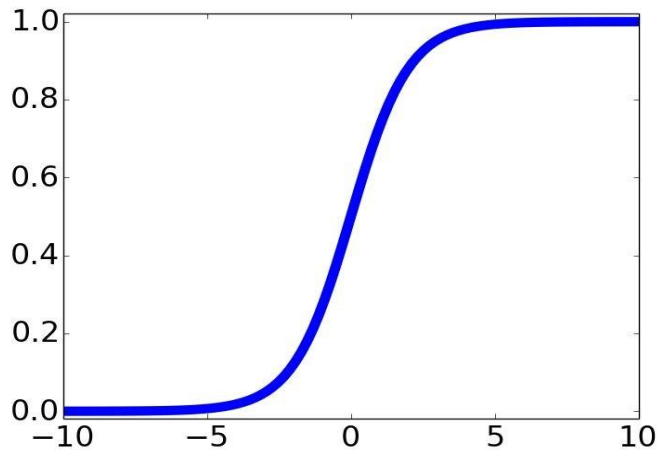
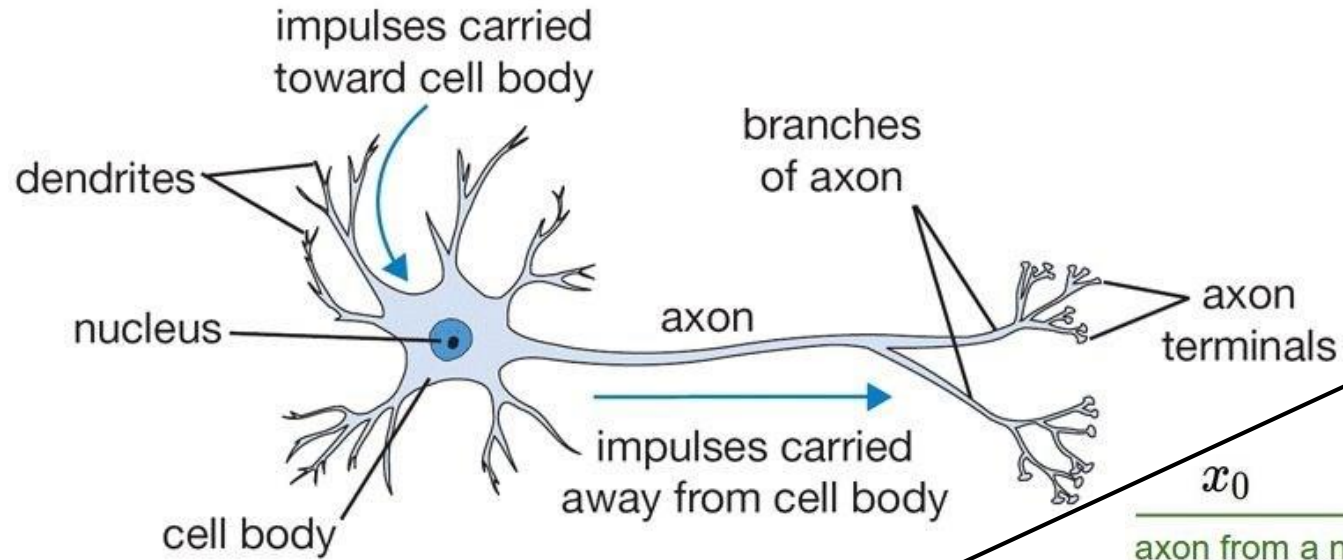
Biological neuron



Artificial neuron

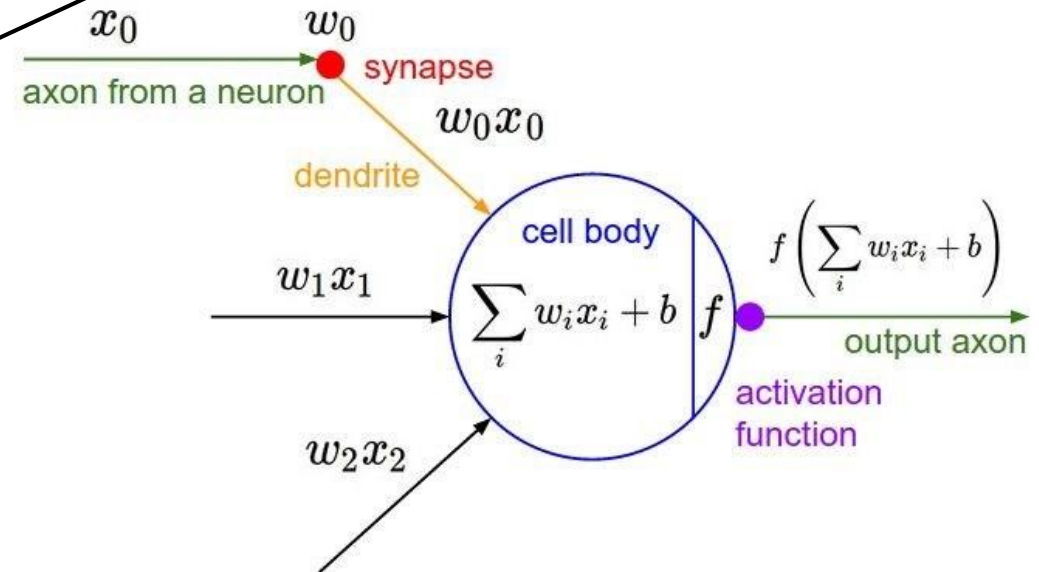


Artificial neuron

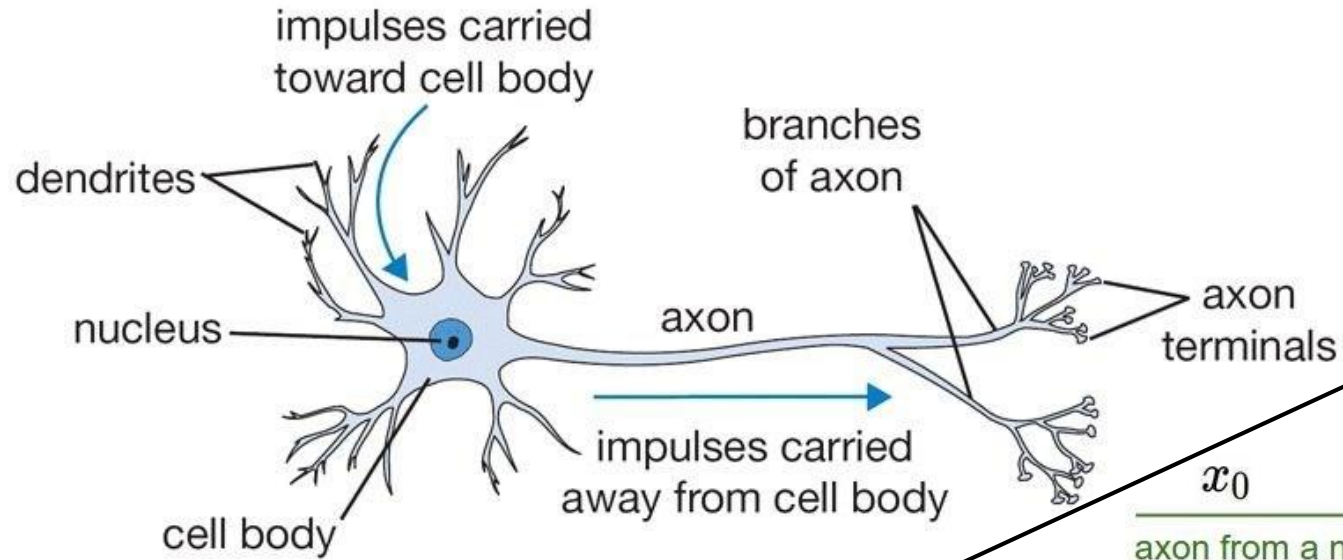


sigmoid activation
function

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



Artificial neuron



```
class Neuron:
```

```
# ...
```

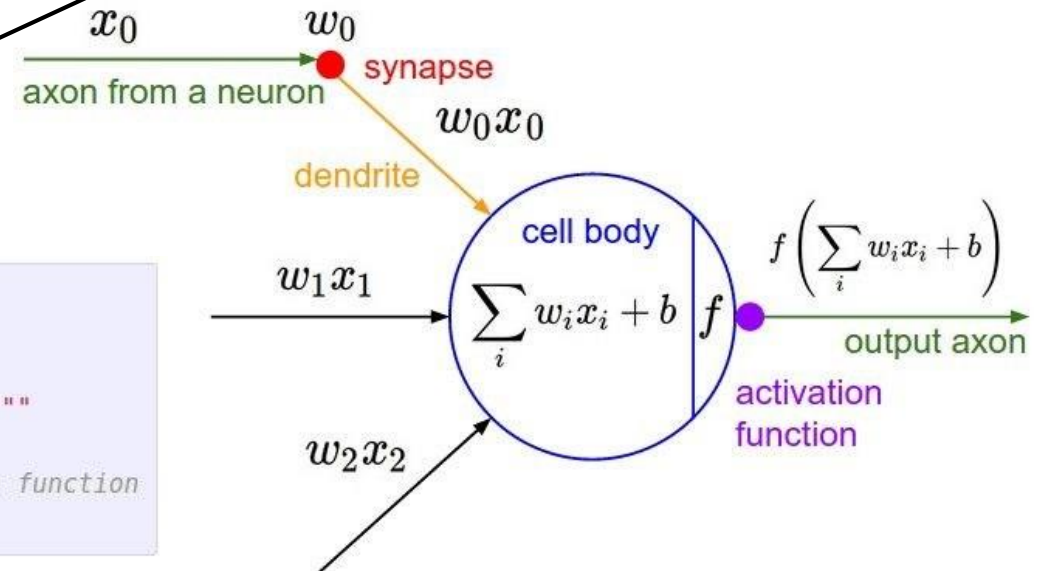
```
def neuron_tick(inputs):
```

```
    """ assume inputs and weights are 1-D numpy arrays and bias is a number """
```

```
    cell_body_sum = np.sum(inputs * self.weights) + self.bias
```

```
    firing_rate = 1.0 / (1.0 + math.exp(-cell_body_sum)) # sigmoid activation function
```

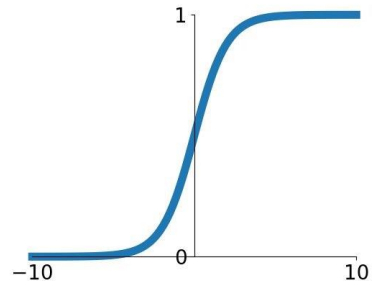
```
    return firing_rate
```



Активационные функции

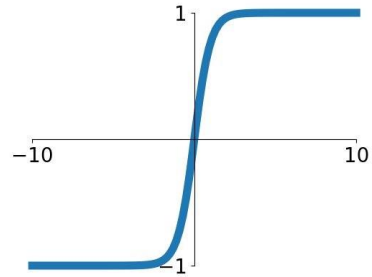
Sigmoid

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



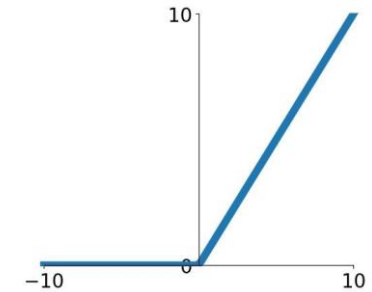
tanh

$$\tanh(x)$$



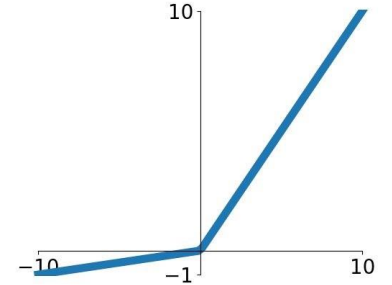
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

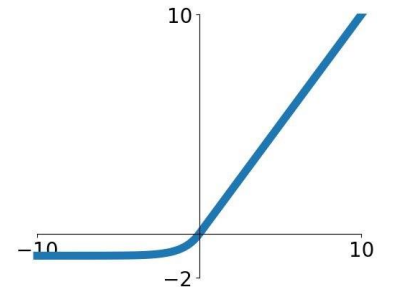


Maxout neuron

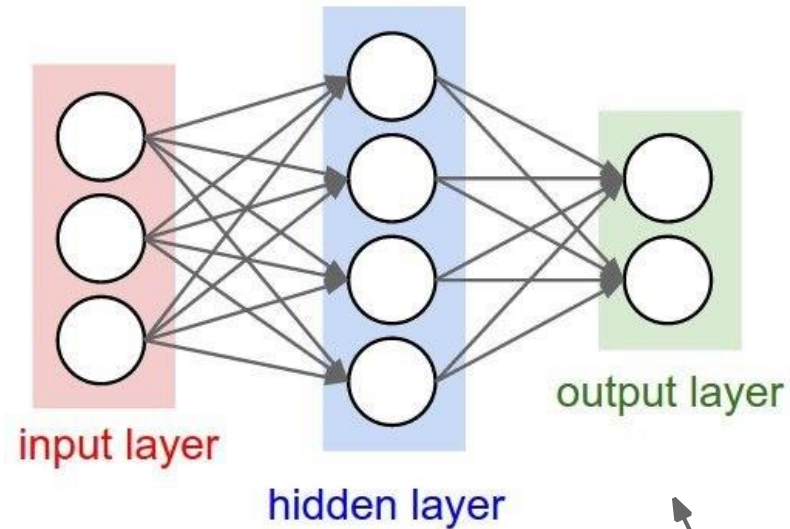
$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

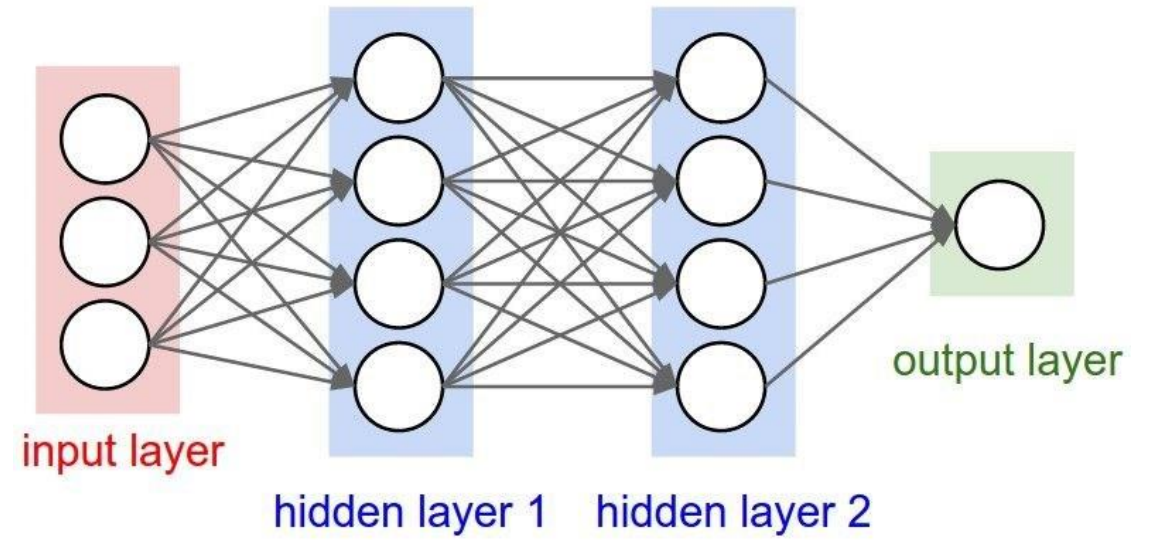
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Neural networks: fully-connected architectures



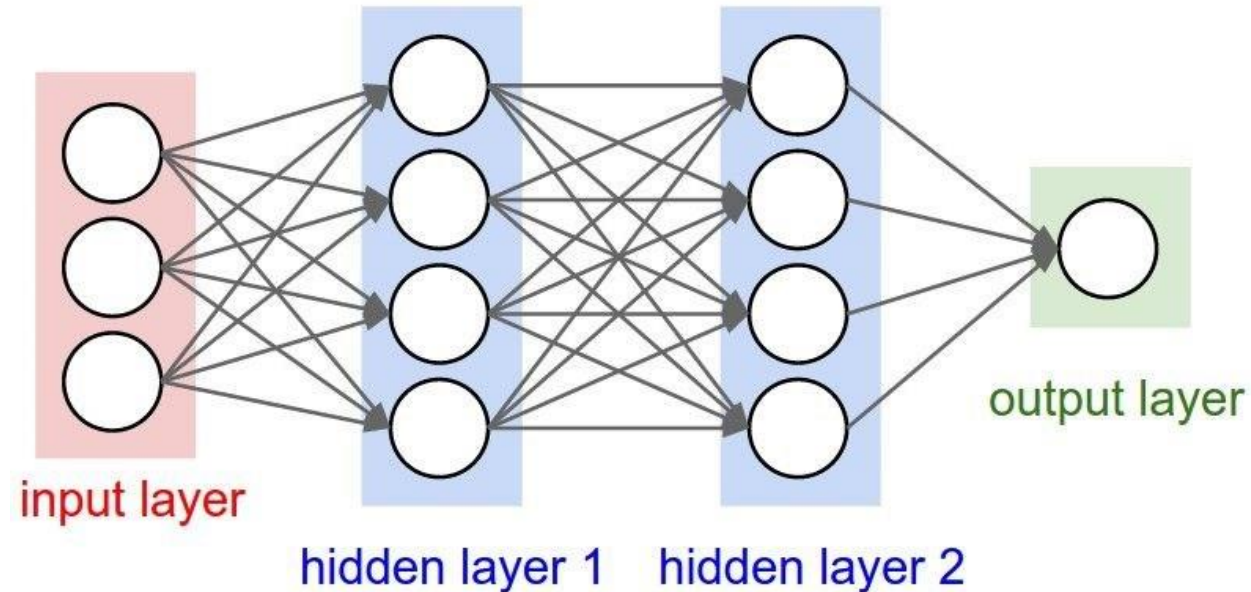
“2-layer Neural Net”, or
“1-hidden-layer Neural Net”



“3-layer Neural Net”, or
“2-hidden-layer Neural Net”

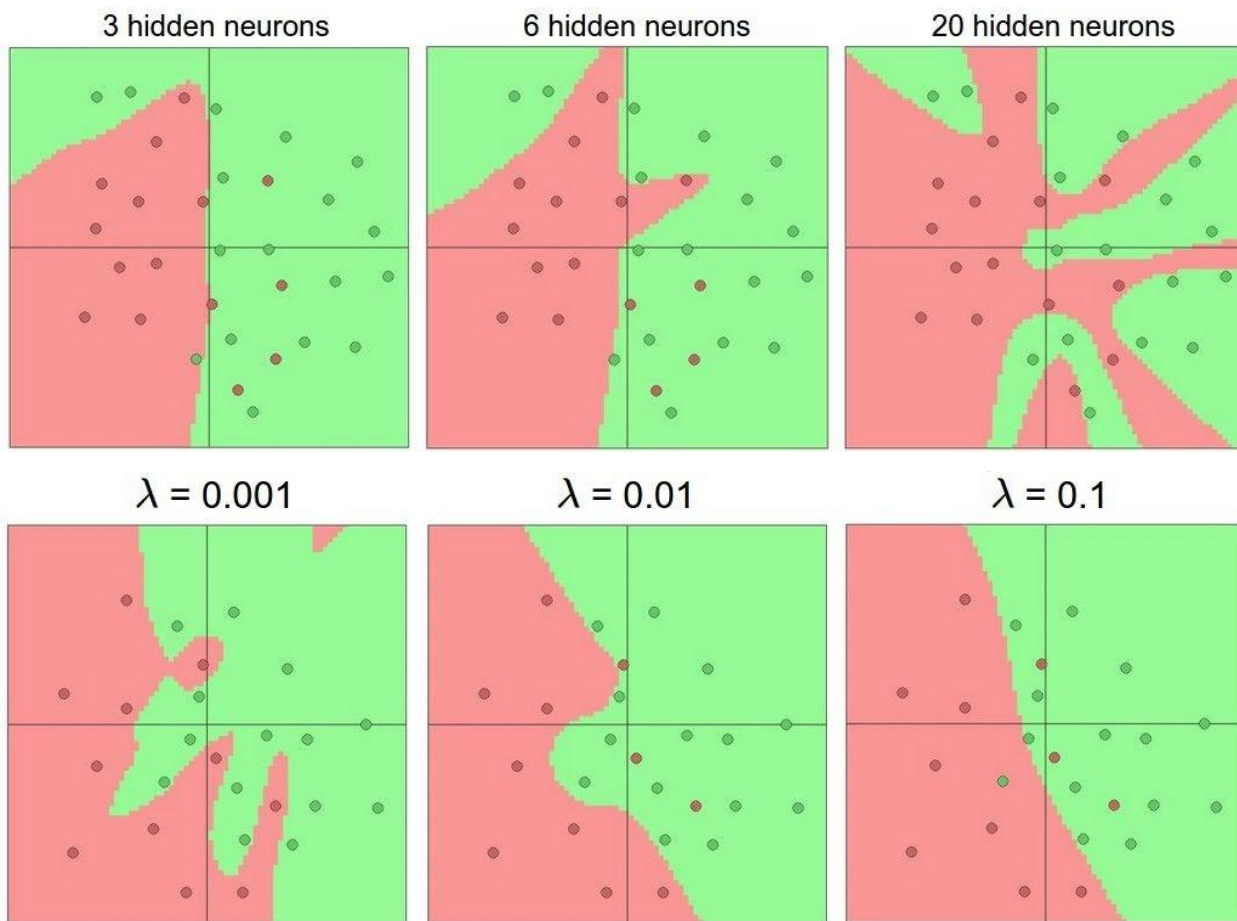
“Fully-connected” layers

Example feed-forward computation of a neural network



```
# forward-pass of a 3-layer neural network:  
f = lambda x: 1.0/(1.0 + np.exp(-x)) # activation function (use sigmoid)  
x = np.random.randn(3, 1) # random input vector of three numbers (3x1)  
h1 = f(np.dot(W1, x) + b1) # calculate first hidden layer activations (4x1)  
h2 = f(np.dot(W2, h1) + b2) # calculate second hidden layer activations (4x1)  
out = np.dot(W3, h2) + b3 # output neuron (1x1)
```

Демо онлайн



Setting the number of layers and their sizes

Setting regularization

<http://cs.stanford.edu/people/karpathy/convnetjs/demo/classify2d.html>

В следующий раз

- Сверточные нейронные сети –
Convolutional Neural Networks (CNN)