

Основы глубинного обучения

Лекция 2

Обратное распространение ошибки. Свёрточные сети.

Евгений Соколов

esokolov@hse.ru

НИУ ВШЭ, 2024

Обучение нейронных сетей

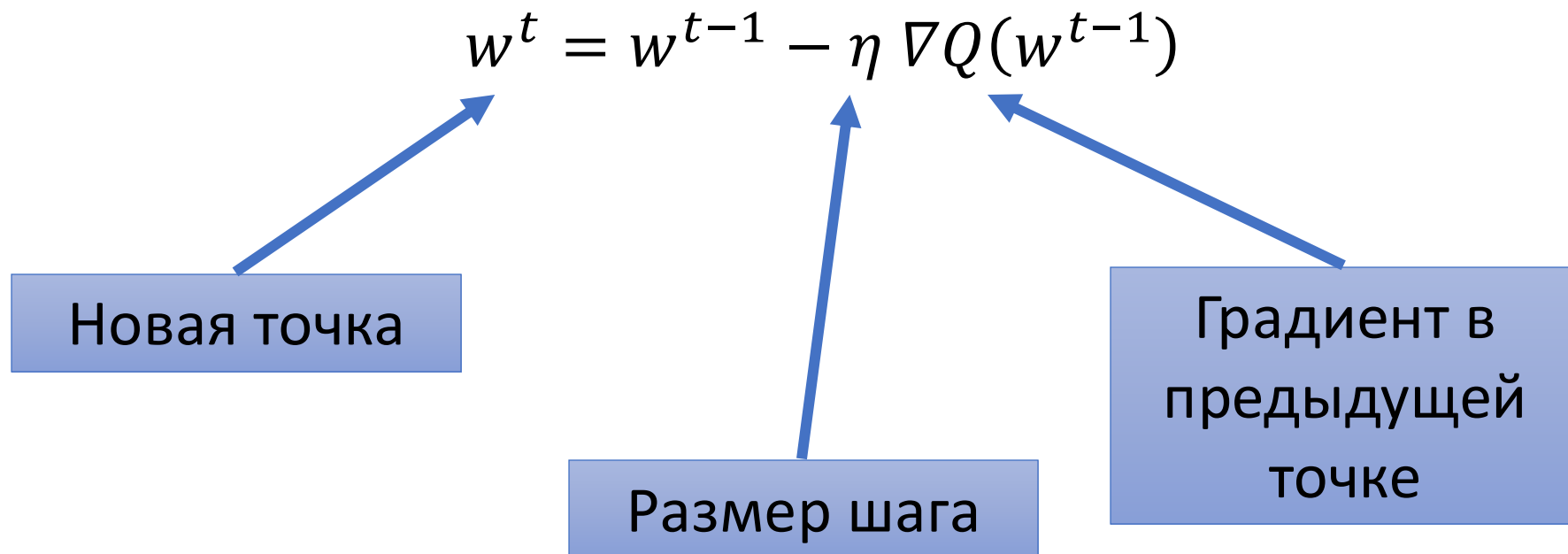
Опрос

Что из этого — формула для шага в градиентном спуске?

1. $w^t = w^{t-1} + \eta \nabla Q(w^t)$
2. $w^t = w^{t-1} - \eta \nabla Q(w^{t-1})$
3. $w^t = w^{t-1} - \eta \nabla Q(w^t)$
4. $w^t = w^{t-1} + \eta \nabla Q(w^0)$

Градиентный спуск

- Повторять до сходимости:



СХОДИМОСТЬ

- Останавливаем процесс, если

$$\|w^t - w^{t-1}\| < \varepsilon$$

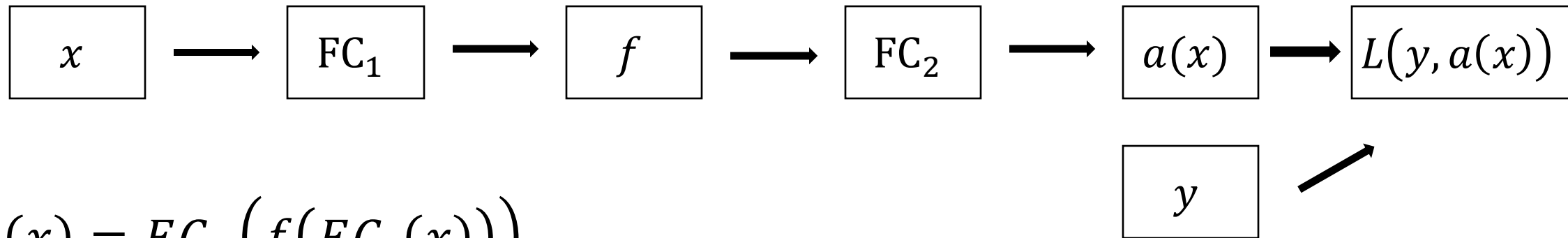
- Другой вариант:

$$\|\nabla Q(w^t)\| < \varepsilon$$

- Обычно в глубинном обучении: останавливаемся, когда ошибка на тестовой выборке перестаёт убывать

Обучение нейронных сетей

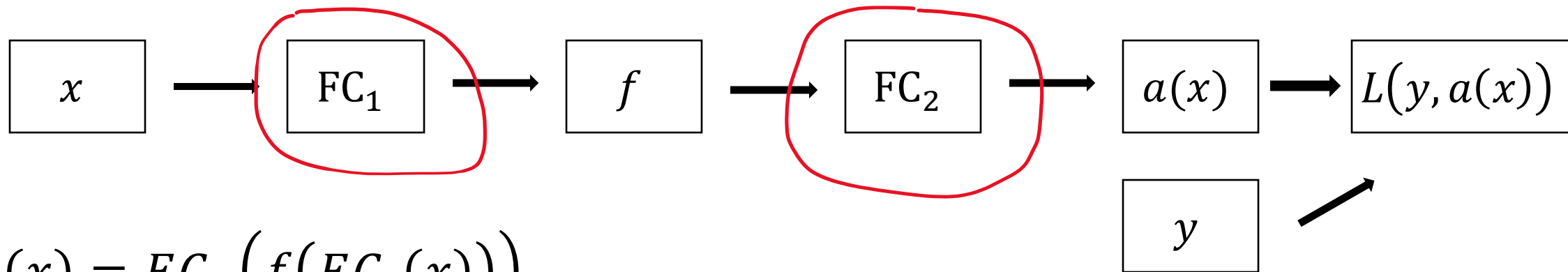
- Все слои обычно дифференцируемы, поэтому можно посчитать производные по всем параметрам



- $a(x) = FC_2 \left(f(FC_1(x)) \right)$
- Где здесь параметры?

Обучение нейронных сетей

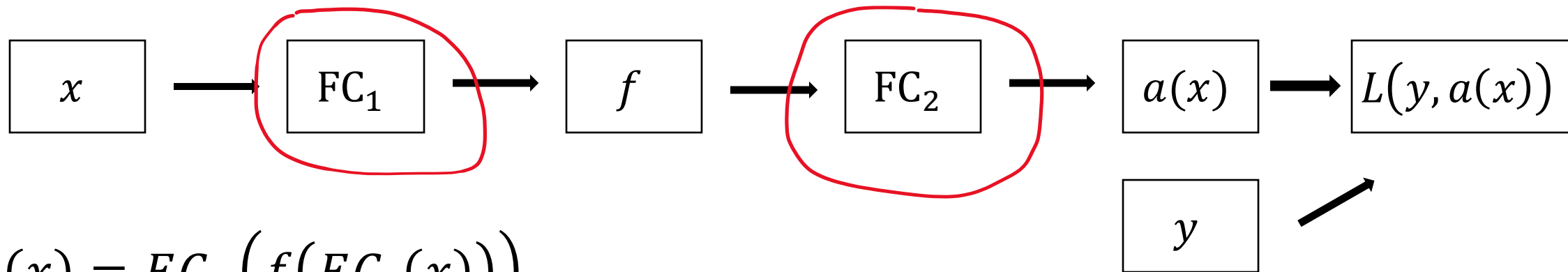
- Все слои обычно дифференцируемы, поэтому можно посчитать производные по всем параметрам



- $a(x) = FC_2 \left(f(FC_1(x)) \right)$
- Где здесь параметры?

Обучение нейронных сетей

- Все слои обычно дифференцируемы, поэтому можно посчитать производные по всем параметрам



- $a(x) = FC_2 \left(f(FC_1(x)) \right)$

$$\frac{1}{\ell} \sum_{i=1}^{\ell} L(y_i, a(x_i)) \rightarrow \min_a$$

Считаем производные

- Для градиентного спуска нужны производные ошибки по параметрам:

$$\frac{\partial}{\partial w_j} L(y_i, a(x_i, w))$$

Считаем производные

- Для градиентного спуска нужны производные ошибки по параметрам:

$$\frac{\partial}{\partial w_j} (a(x_i, w) - y_i)^2$$

Считаем производные

- Для градиентного спуска нужны производные ошибки по параметрам:

$$\frac{\partial}{\partial w_j} (a(x_i, w) - y_i)^2 = 2(a(x_i, w) - y_i) \frac{\partial}{\partial w_j} a(x_i, w)$$

как сильно изменится
ошибка, если пошевелить w_j ? как сильно изменится
ошибка, если пошевелить
 $a(x_i, w)$? как сильно изменится
 $a(x_i, w)$, если
пошевелить w_j ?

Считаем производные

$$\frac{\partial}{\partial w_j} (a(x_i, w) - y_i)^2 = 2(a(x_i, w) - y_i) \frac{\partial}{\partial w_j} a(x_i, w)$$

как сильно изменится
ошибка, если пошевелить w_j ?

как сильно изменится
ошибка, если пошевелить
 $a(x_i, w)$?

как сильно изменится
 $a(x_i, w)$, если
пошевелить w_j ?

- $a(x_i, w) = 10, y_i = 9.99$: $2 * 0.01 * \frac{\partial}{\partial w_j} a(x_i, w)$
- $a(x_i, w) = 10, y_i = 1$: $2 * 9 * \frac{\partial}{\partial w_j} a(x_i, w)$

Считаем производные

- Для градиентного спуска нужны производные ошибки по параметрам:

$$\frac{\partial}{\partial w_j} L(y_i, a(x_i, w)) = \frac{\partial}{\partial z} L(y_i, z) \Big|_{z=a(x_i, w)} \frac{\partial}{\partial w_j} a(x_i, w)$$

как сильно изменится
ошибка, если пошевелить w_j ?

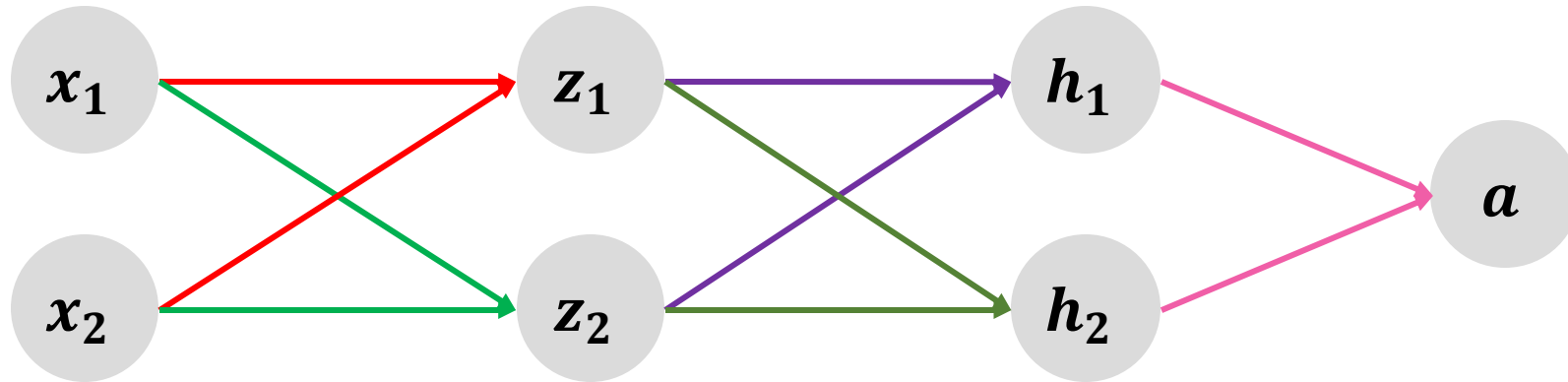
как сильно изменится
ошибка, если пошевелить
 $a(x_i, w)$?

как сильно изменится
 $a(x_i, w)$, если
пошевелить w_j ?

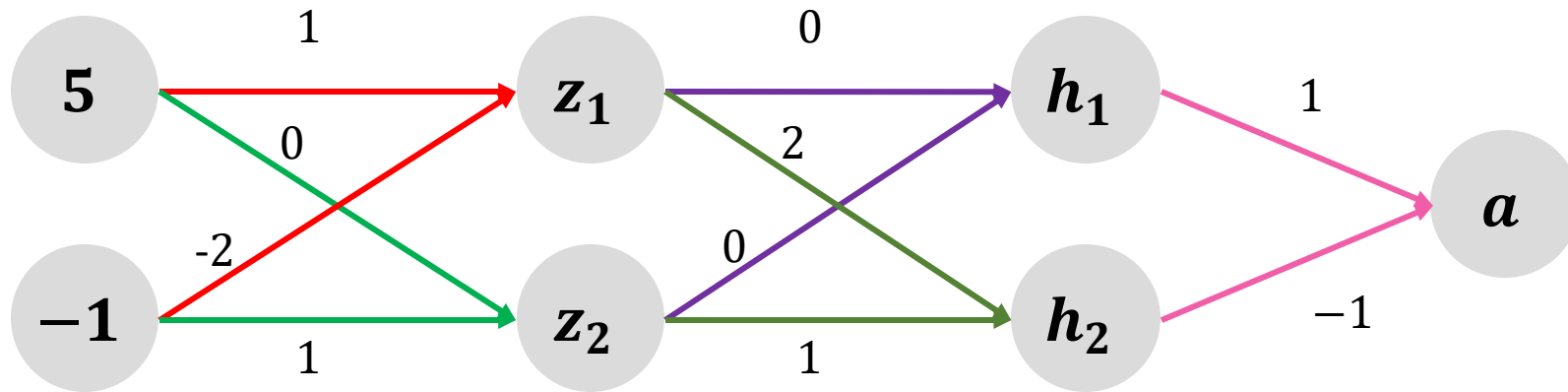
Считаем производные

- Следующая задача — научиться вычислять $\frac{\partial}{\partial w_j} a(x_i, w)$

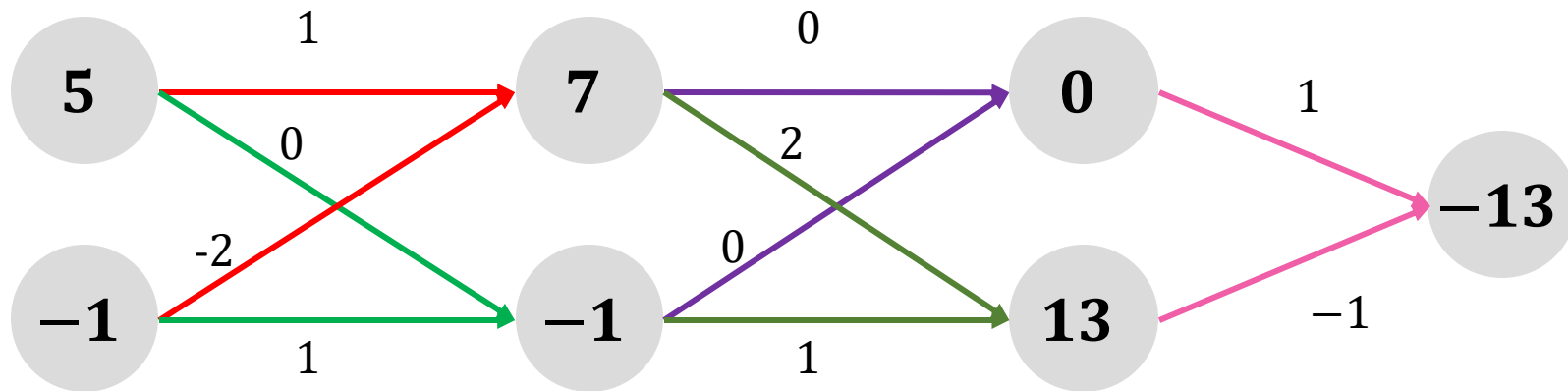
Как считать производные?



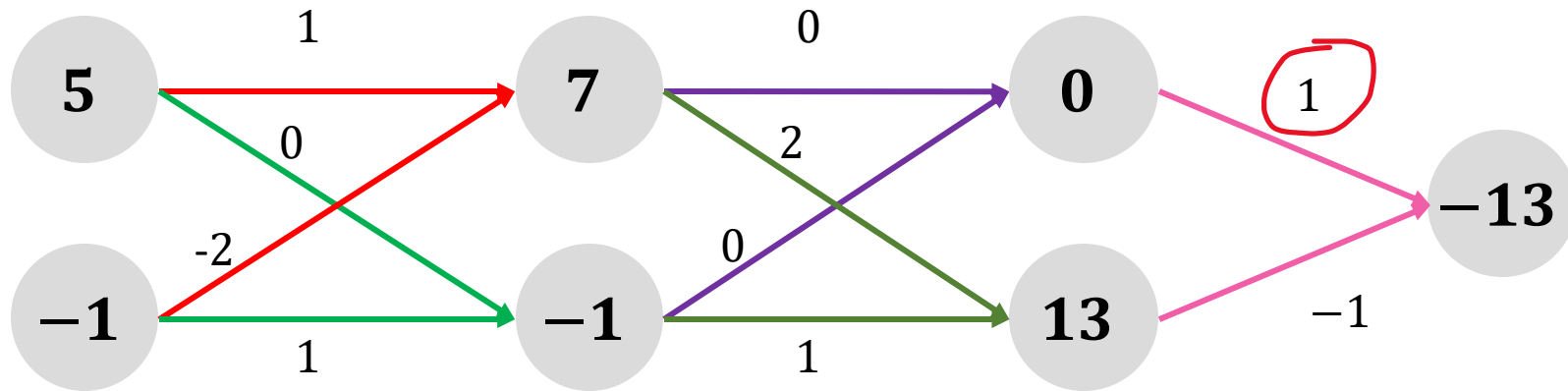
Как считать производные?



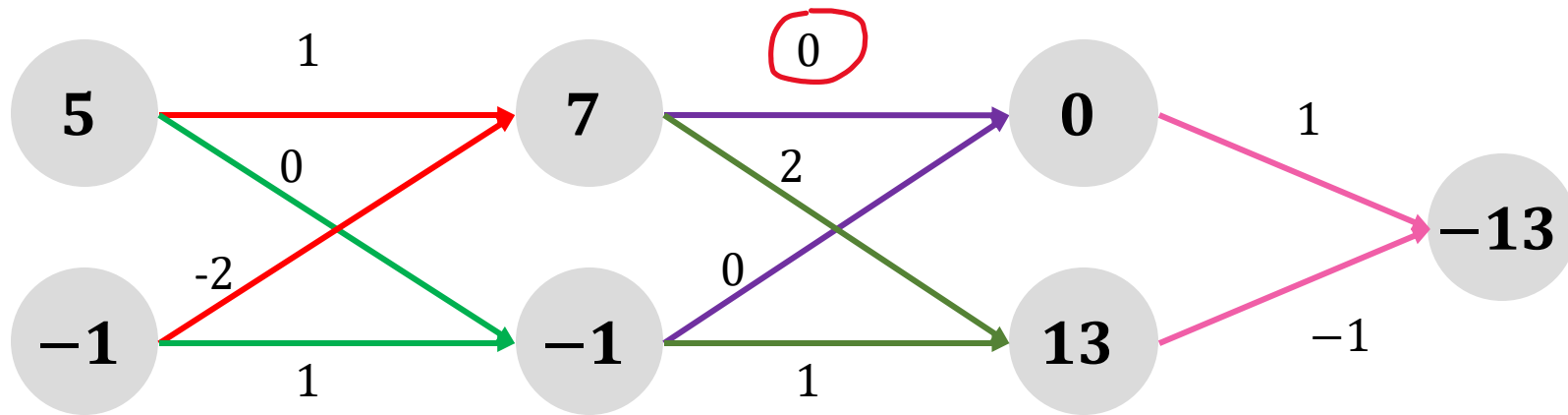
Как считать производные?



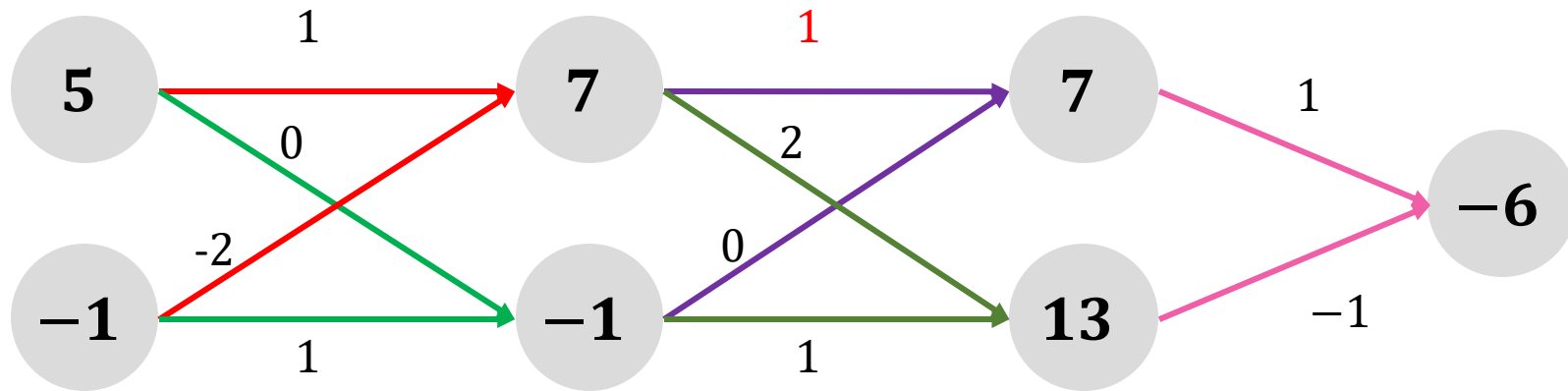
Как считать производные?



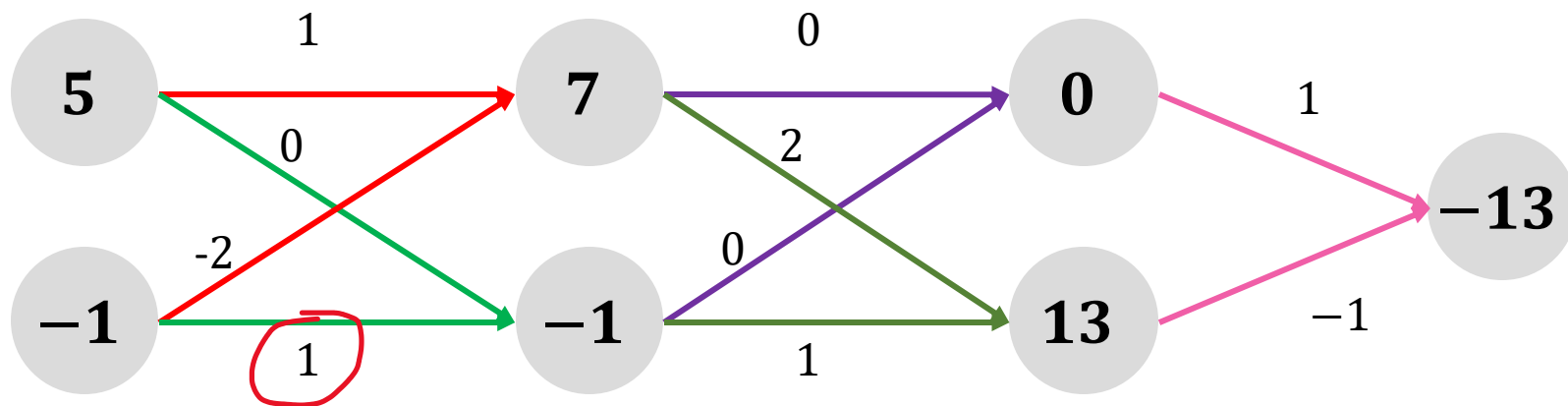
Как считать производные?



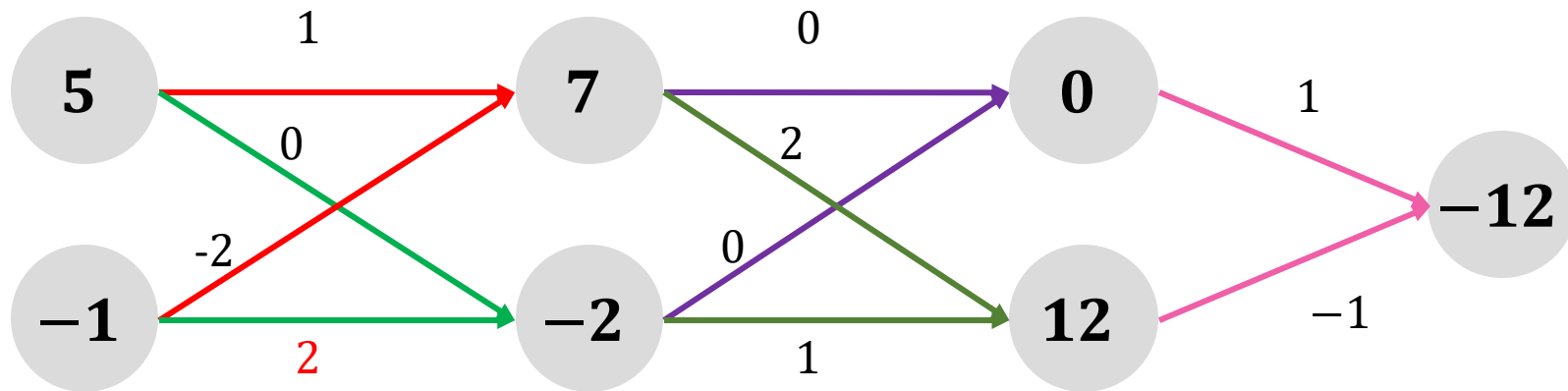
Как считать производные?



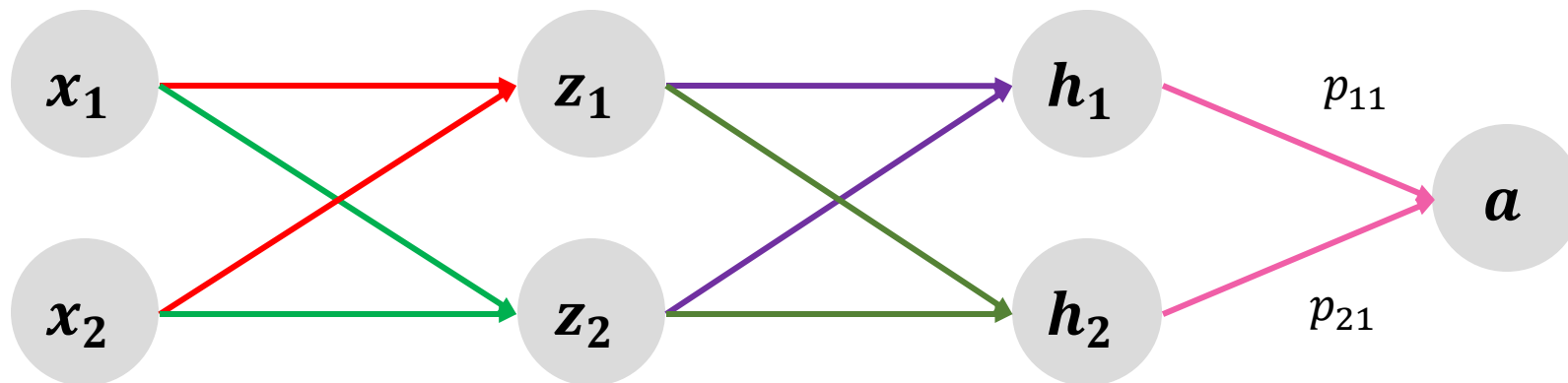
Как считать производные?



Как считать производные?



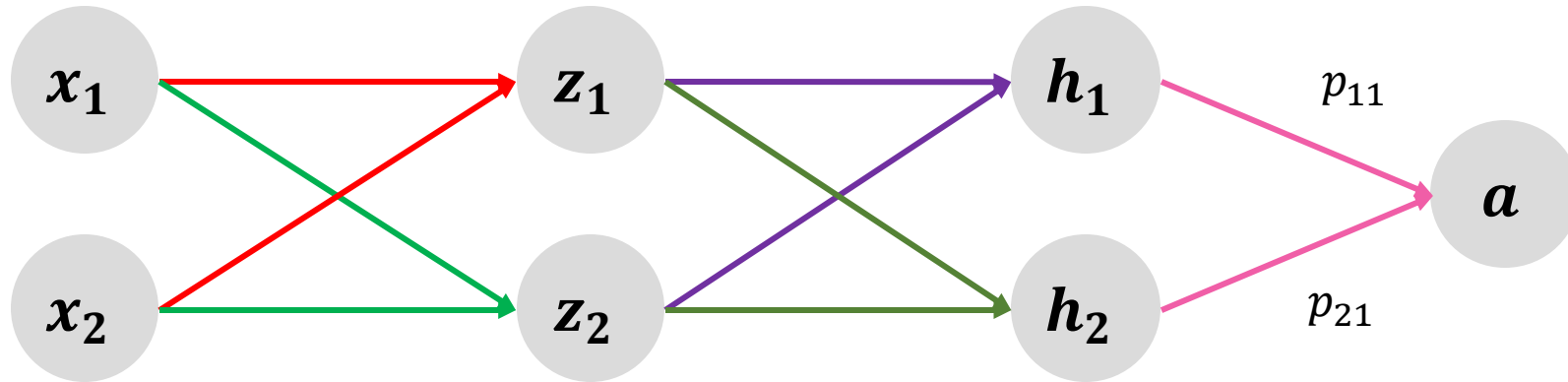
Как считать производные?



$$a(x) = p_{11}h_1(x) + p_{21}h_2(x)$$

$$\frac{\partial a}{\partial p_{11}} = ?$$

Как считать производные?

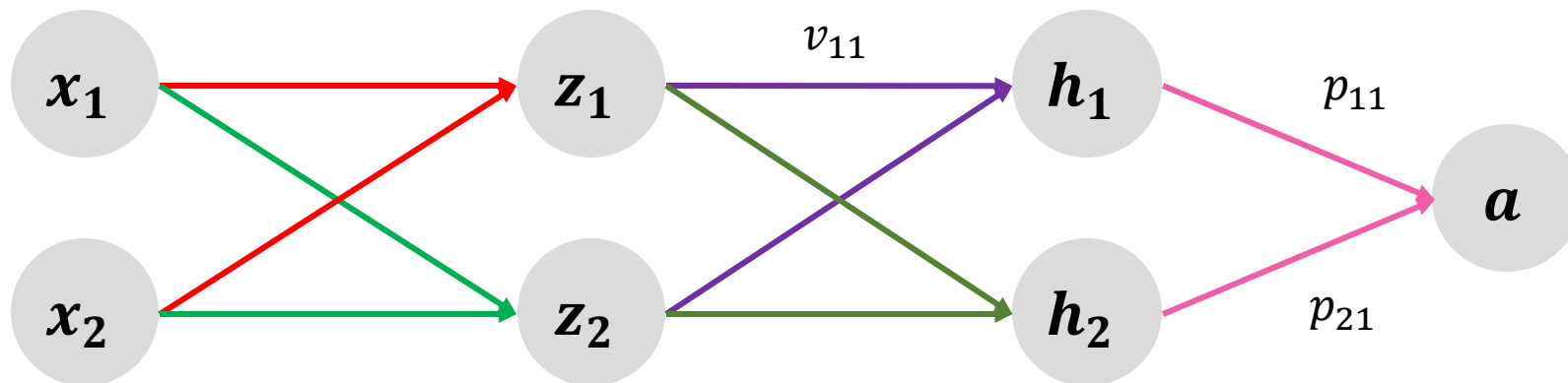


$$a(x) = p_{11}h_1(x) + p_{21}h_2(x)$$

$$\frac{\partial a}{\partial p_{11}} = h_1(x)$$

- Чем больше $h_1(x)$, тем сильнее p_{11} влияет на a

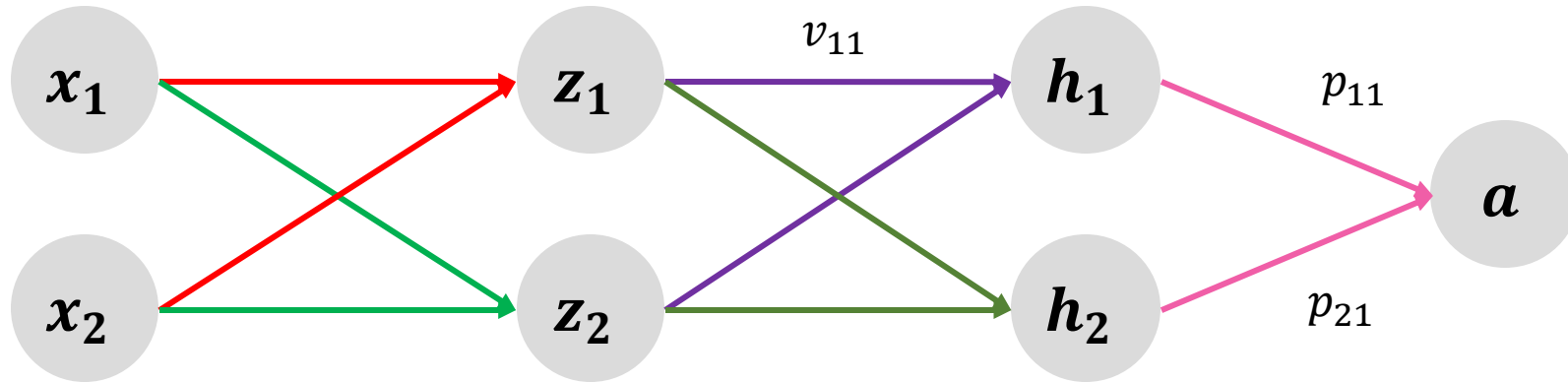
Как считать производные?



$$a(x) = p_{11}f(v_{11}z_1(x) + v_{21}z_2(x)) + p_{21}h_2(x)$$

$$\frac{\partial a}{\partial v_{11}} = ?$$

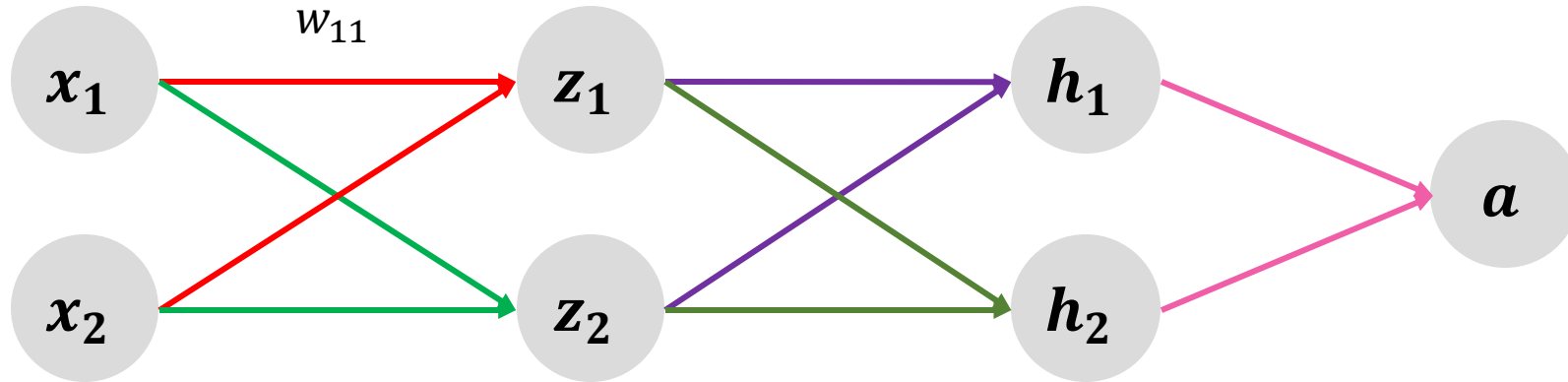
Как считать производные?



$$a(x) = p_{11}f(v_{11}z_1(x) + v_{21}z_2(x)) + p_{21}h_2(x)$$

$$\frac{\partial a}{\partial v_{11}} = \frac{\partial a}{\partial h_1} \frac{\partial h_1}{\partial v_{11}}$$

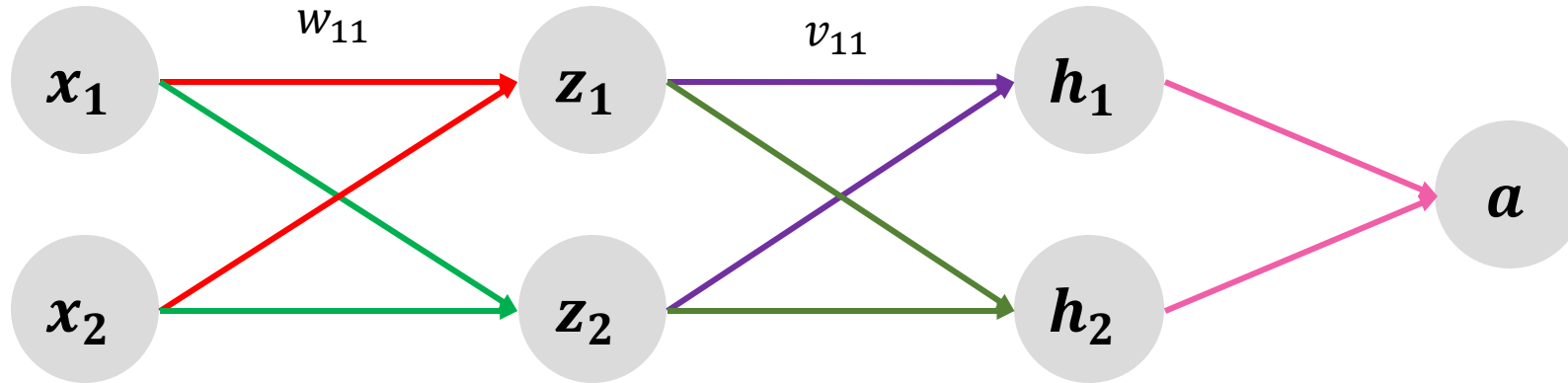
Как считать производные?



$$\frac{\partial a}{\partial w_{11}} = ?$$

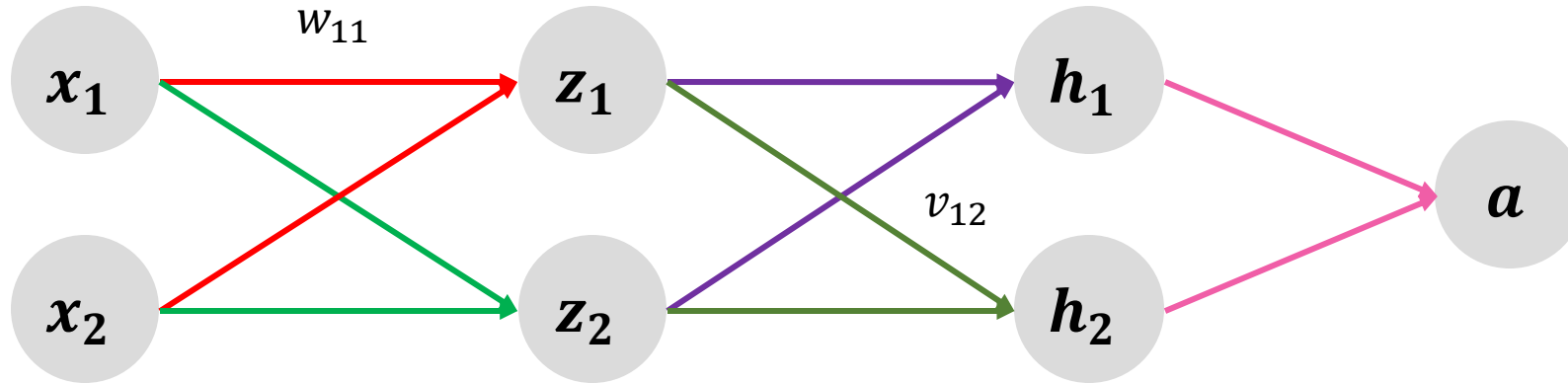
- Показывает, как сильно изменится a при изменении w_{11}

Как считать производные?



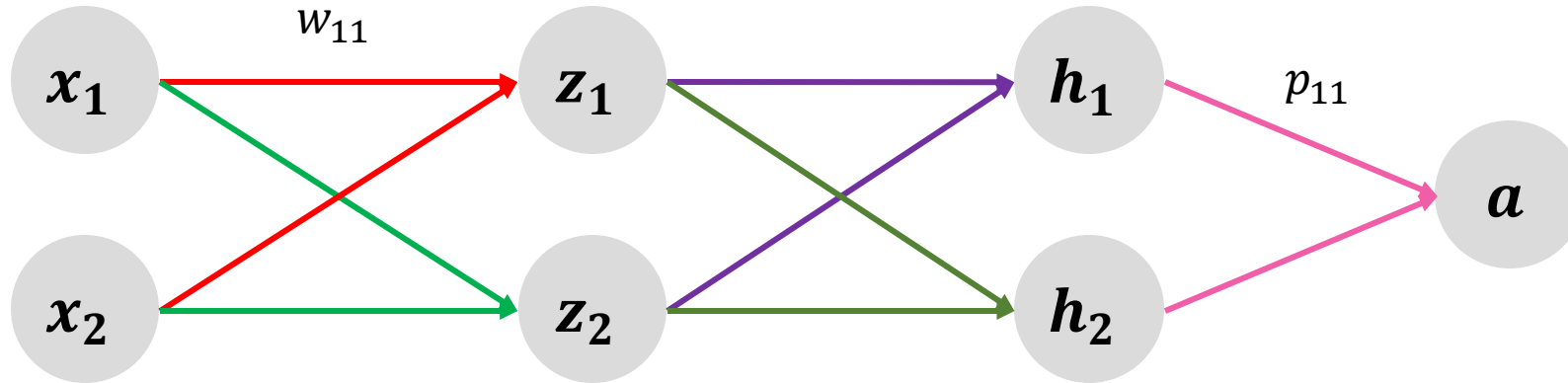
- Как сильно изменится a при изменении w_{11} ?
- Влияет ли на это v_{11} ?

Как считать производные?



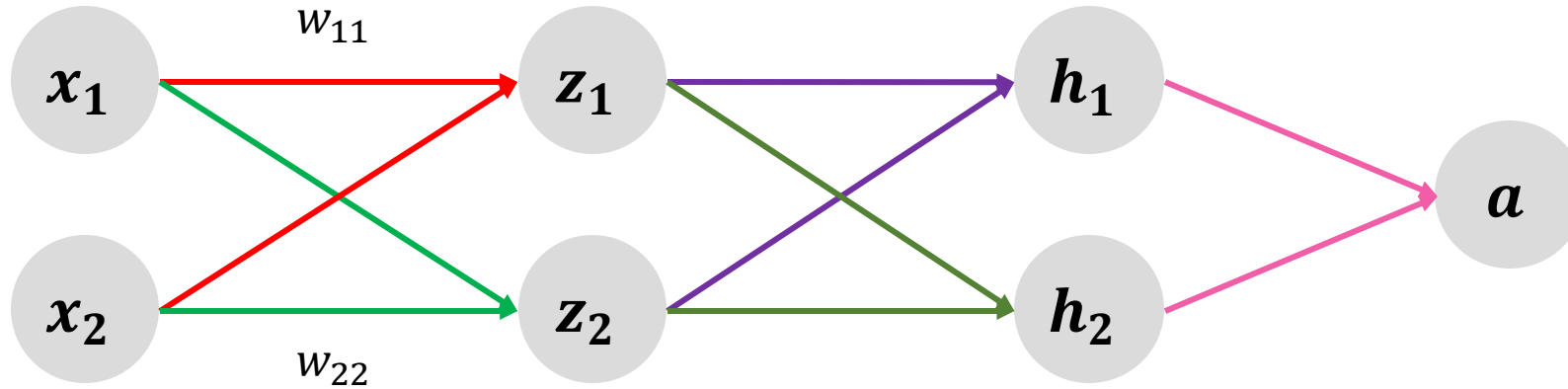
- Как сильно изменится a при изменении w_{11} ?
- Влияет ли на это v_{12} ?

Как считать производные?



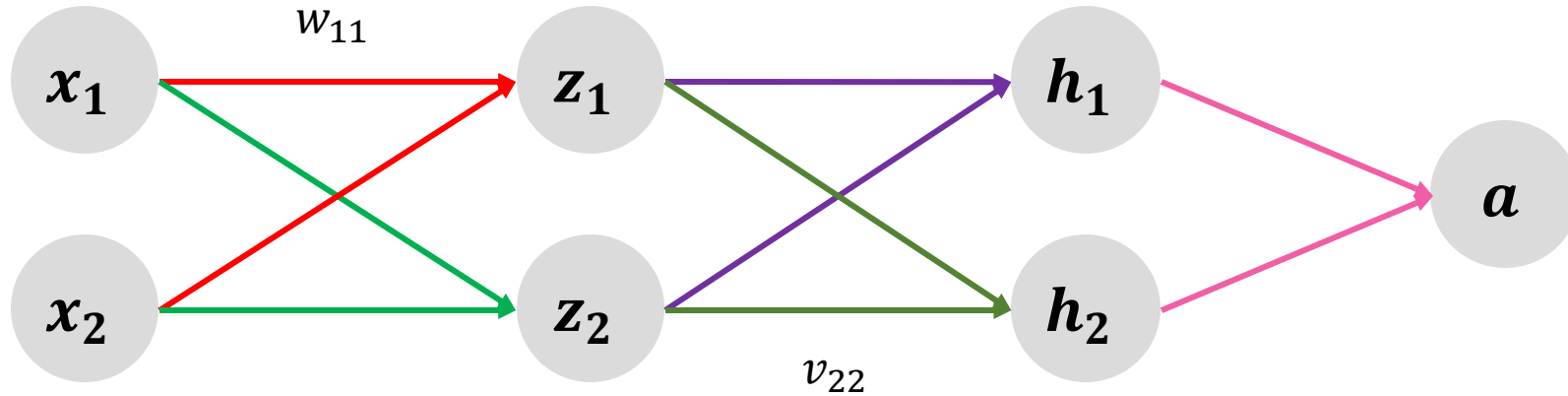
- Как сильно изменится a при изменении w_{11} ?
- Влияет ли на это p_{11} ?

Как считать производные?



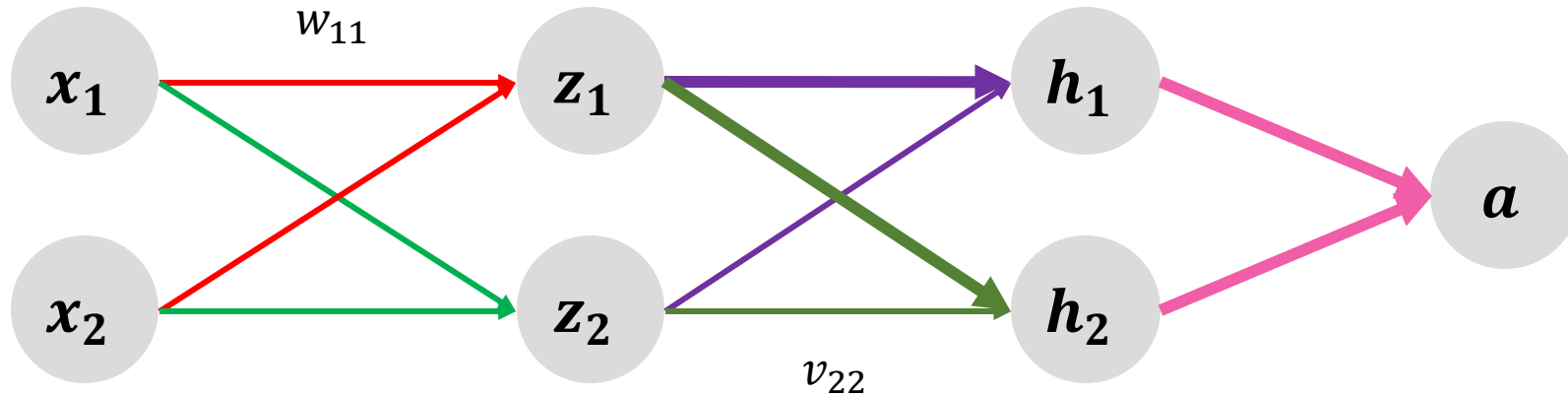
- Как сильно изменится a при изменении w_{11} ?
- Влияет ли на это w_{22} ?

Как считать производные?



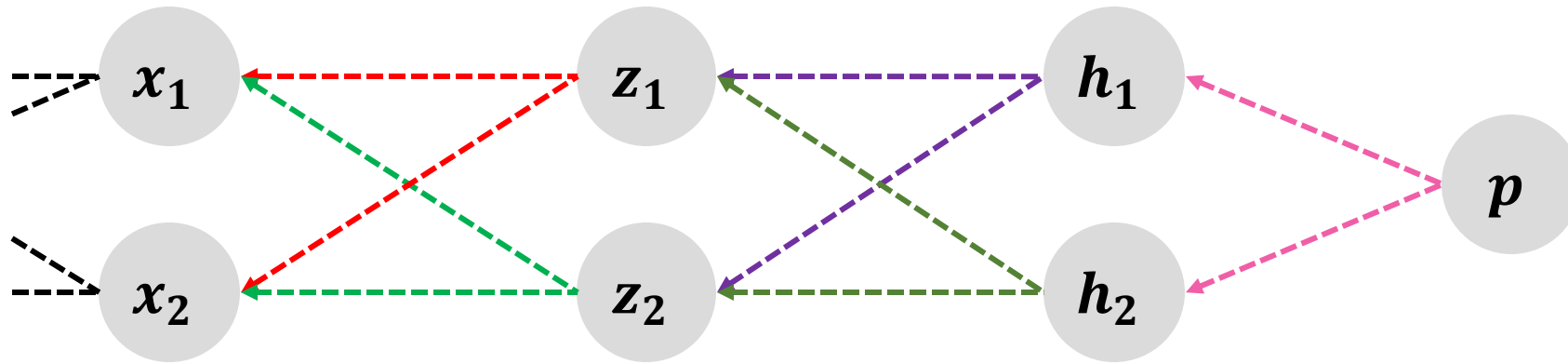
- Как сильно изменится a при изменении w_{11} ?
- Влияет ли на это v_{22} ?

Как считать производные?



$$\frac{\partial a}{\partial w_{11}} = \frac{\partial a}{\partial h_1} \frac{\partial h_1}{\partial z_1} \frac{\partial z_1}{\partial w_{11}} + \frac{\partial a}{\partial h_2} \frac{\partial h_2}{\partial z_1} \frac{\partial z_1}{\partial w_{11}}$$

Как считать производные?



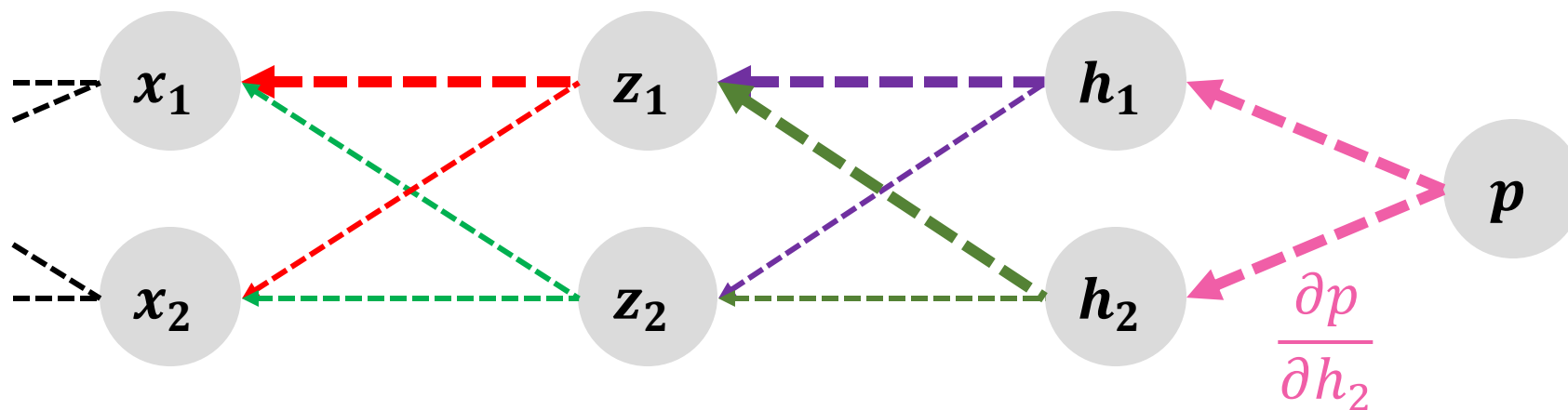
- Мы как бы идём в обратную сторону по графу и считаем производные
- Метод обратного распространения ошибки (backpropagation)

$$3: \quad \frac{\partial p}{\partial h_1} \quad \frac{\partial p}{\partial h_2}$$

$$2: \quad \frac{\partial p}{\partial z_1} = \frac{\partial p}{\partial h_1} \frac{\partial h_1}{\partial z_1} + \frac{\partial p}{\partial h_2} \frac{\partial h_2}{\partial z_1} \quad \frac{\partial p}{\partial z_2} = \frac{\partial p}{\partial h_1} \frac{\partial h_1}{\partial z_2} + \frac{\partial p}{\partial h_2} \frac{\partial h_2}{\partial z_2}$$

$$1: \quad \frac{\partial p}{\partial x_1} = \frac{\partial p}{\partial h_1} \frac{\partial h_1}{\partial z_1} \frac{\partial z_1}{\partial x_1} + \frac{\partial p}{\partial h_2} \frac{\partial h_2}{\partial z_1} \frac{\partial z_1}{\partial x_1} + \frac{\partial p}{\partial h_1} \frac{\partial h_1}{\partial z_2} \frac{\partial z_2}{\partial x_1} + \frac{\partial p}{\partial h_2} \frac{\partial h_2}{\partial z_2} \frac{\partial z_2}{\partial x_1}$$

$$\frac{\partial p}{\partial x_2} = \frac{\partial p}{\partial h_1} \frac{\partial h_1}{\partial z_1} \frac{\partial z_1}{\partial x_2} + \frac{\partial p}{\partial h_2} \frac{\partial h_2}{\partial z_1} \frac{\partial z_1}{\partial x_2} + \frac{\partial p}{\partial h_1} \frac{\partial h_1}{\partial z_2} \frac{\partial z_2}{\partial x_2} + \frac{\partial p}{\partial h_2} \frac{\partial h_2}{\partial z_2} \frac{\partial z_2}{\partial x_2}$$



Backprop

- Во многие формулы входят одни и те же производные
- В backprop каждая частная производная вычисляется один раз — вычисление производных по слою N сводится к перемножению матрицы производных по слою $N+1$ и некоторых векторов

Полносвязные сети для
изображений

MNIST

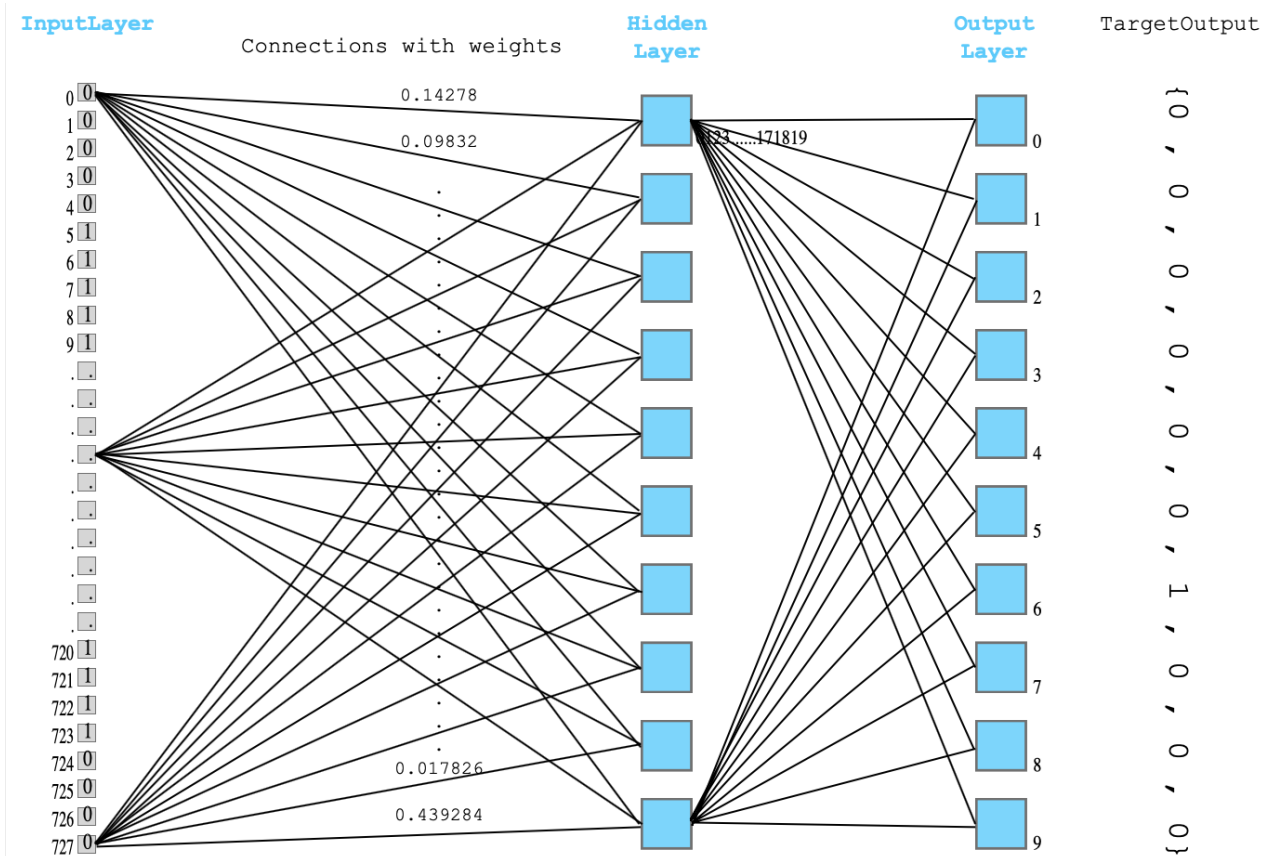


MNIST

- Изображения 28 x 28
- Изображения центрированы
- 60.000 объектов в обучающей выборке

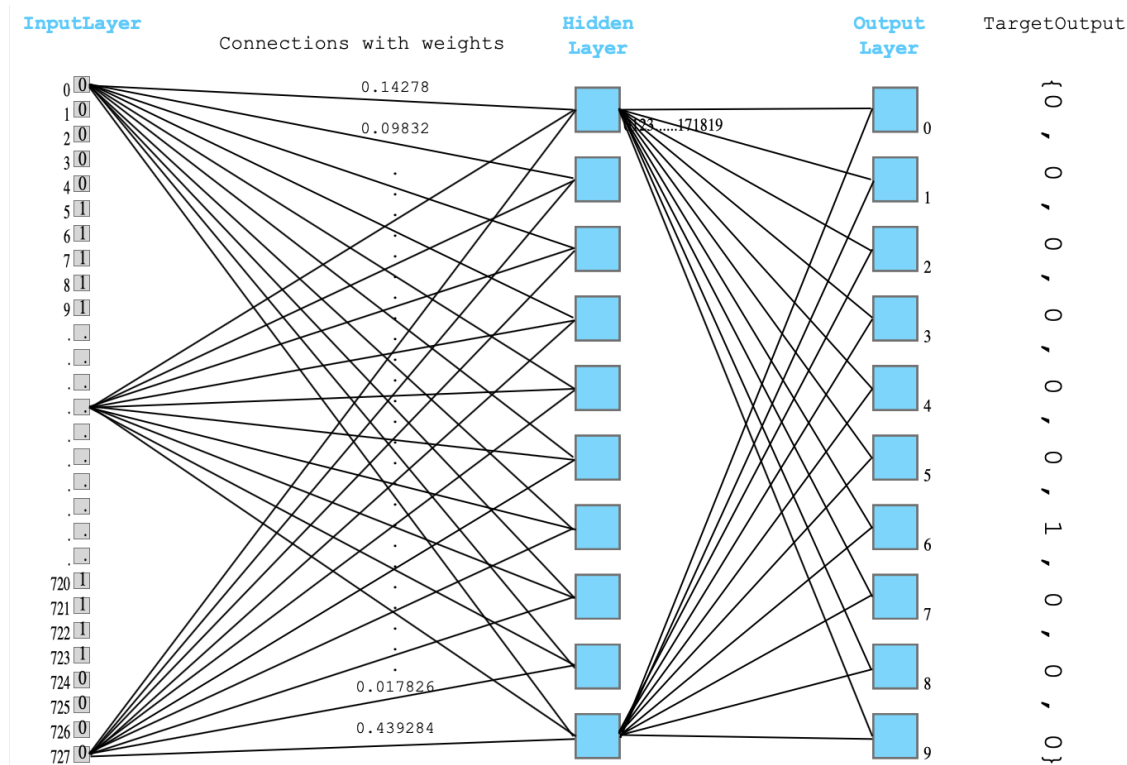
MNIST

- Что может выучить полносвязная сеть?

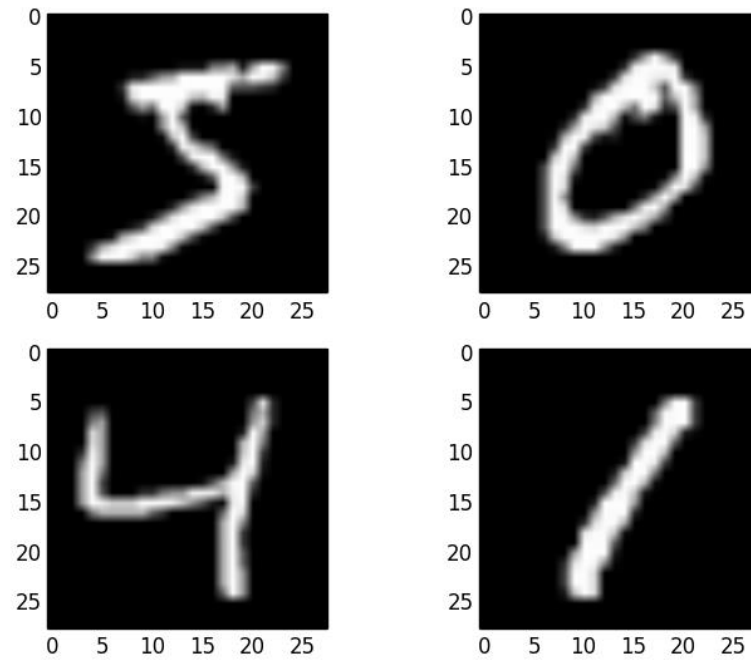


MNIST

- Каждый нейрон может детектировать заполненность конкретного набора пикселей

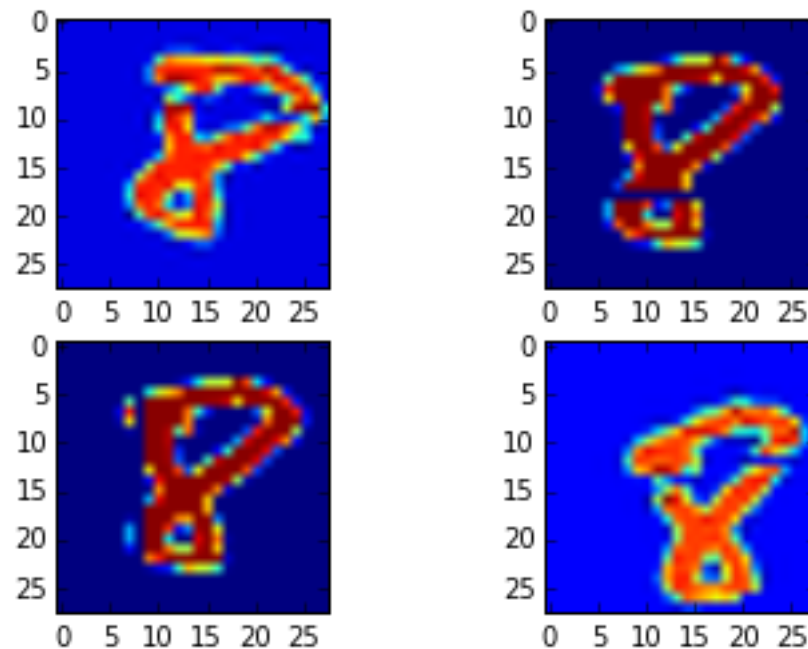


MNIST



MNIST

- Если немного сдвинуть цифру, то нейрон уже не будет на неё реагировать



Число параметров

- 784 входа
- Полносвязный слой: 1000 нейронов
- Выходной слой: 10 нейронов (по одному на каждый класс)
- Весов между входным и полносвязным слоями:

$$(784 + 1) * 1000 = 785.000$$

- Весов между полносвязным и выходным слоями:

$$(1000 + 1) * 10 = 10.010$$

Число параметров

- Можно добиться хорошего качества полносвязными сетями (с аугментацией)
- <https://arxiv.org/abs/1003.0358>

Table 1: Error rates on MNIST test set.

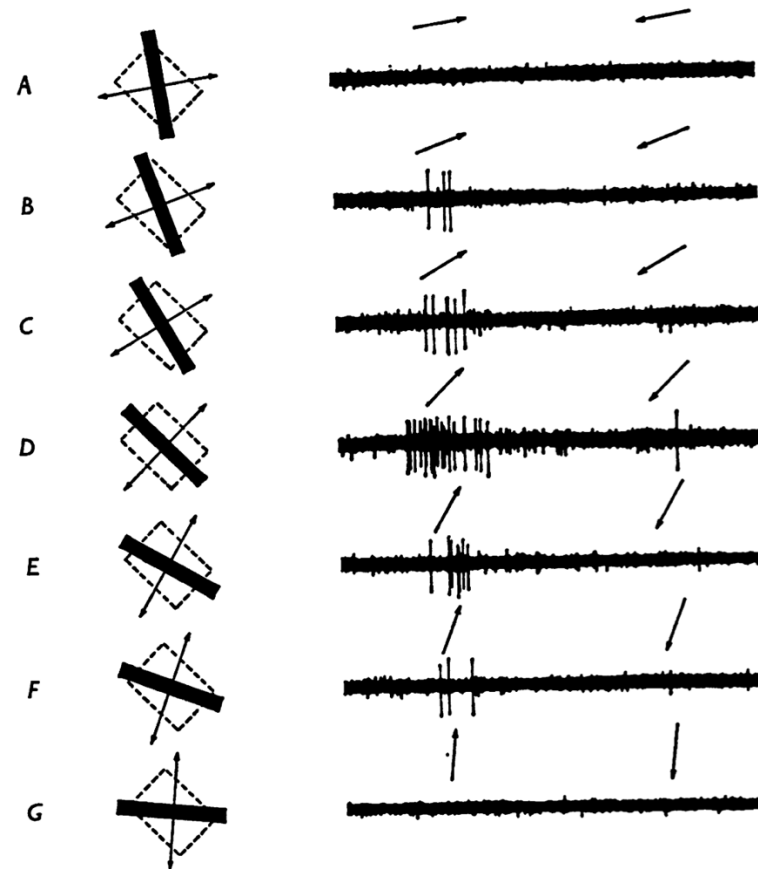
ID	architecture (number of neurons in each layer)	test error for best validation [%]	best test error [%]	simulation time [h]	weights [milions]
1	1000, 500, 10	0.49	0.44	23.4	1.34
2	1500, 1000, 500, 10	0.46	0.40	44.2	3.26
3	2000, 1500, 1000, 500, 10	0.41	0.39	66.7	6.69
4	2500, 2000, 1500, 1000, 500, 10	0.35	0.32	114.5	12.11
5	9×1000 , 10	0.44	0.43	107.7	8.86

Полносвязные слои для изображений

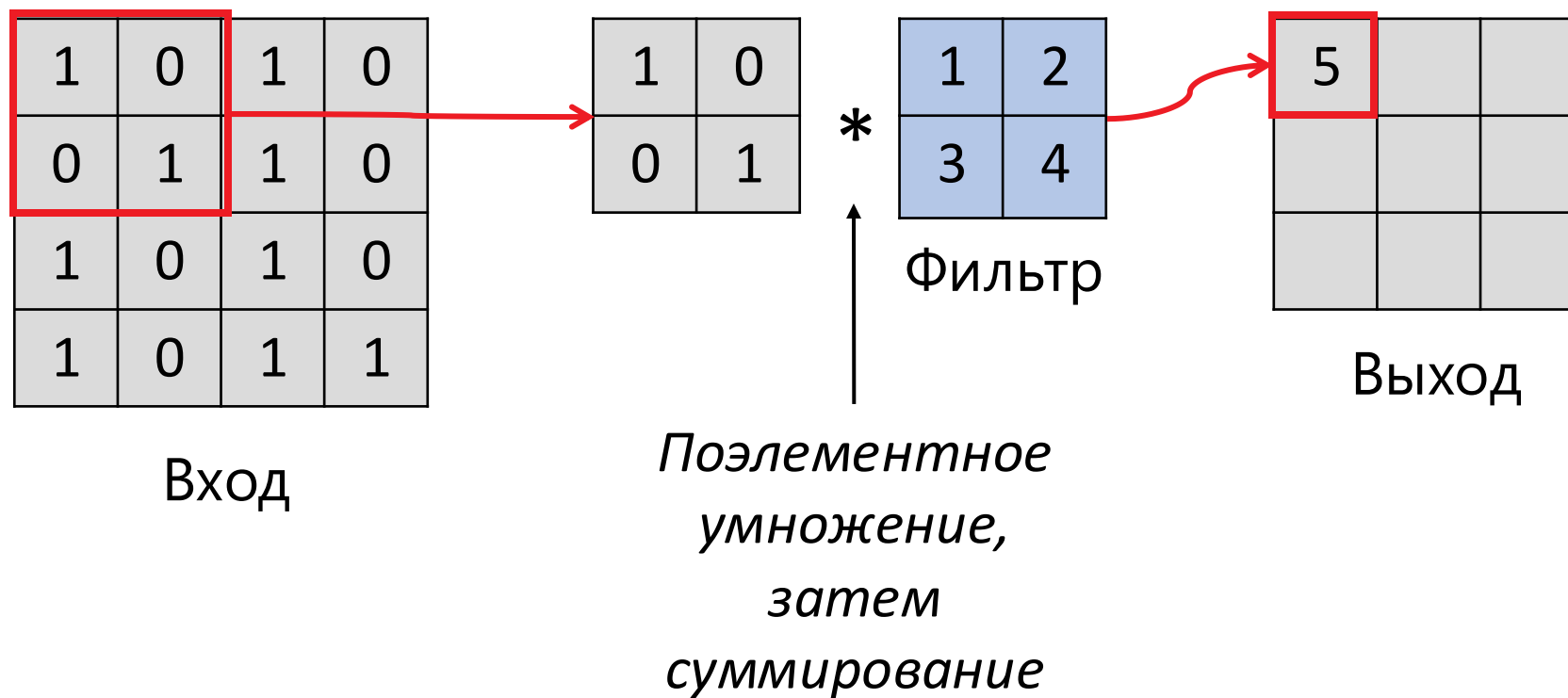
- Очень много параметров
- Легко могут переобучиться
- Не учитывают специфику изображений (сдвиги, небольшие изменения формы и т.д.)
- Один из лучших способов борьбы с переобучением — снижение числа параметров

Свёртки

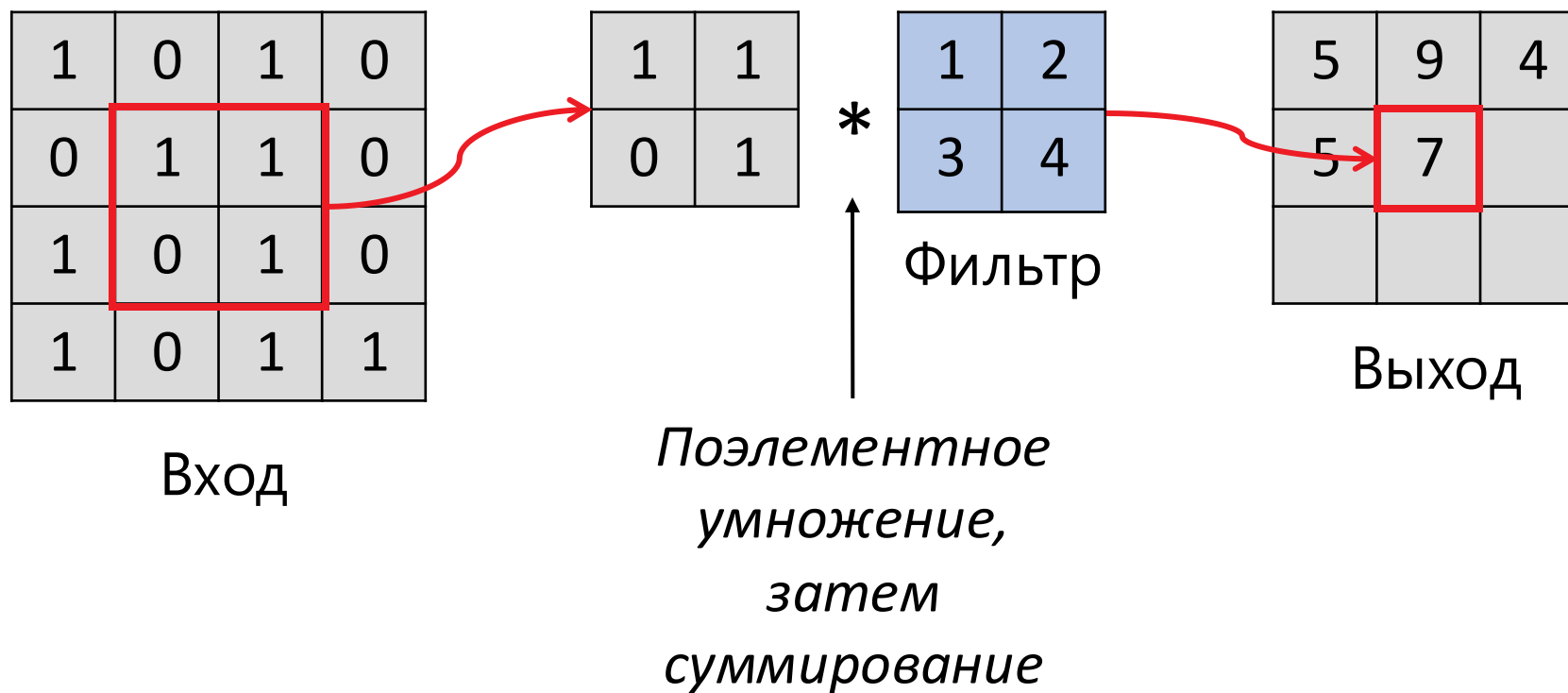
Эксперименты со зрительной корой



Свёртка



Свёртка



Свёртка

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \boxed{2}$$

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \boxed{2}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \boxed{1}$$

$$\begin{bmatrix} 0 & 2 \\ 3 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \boxed{0}$$

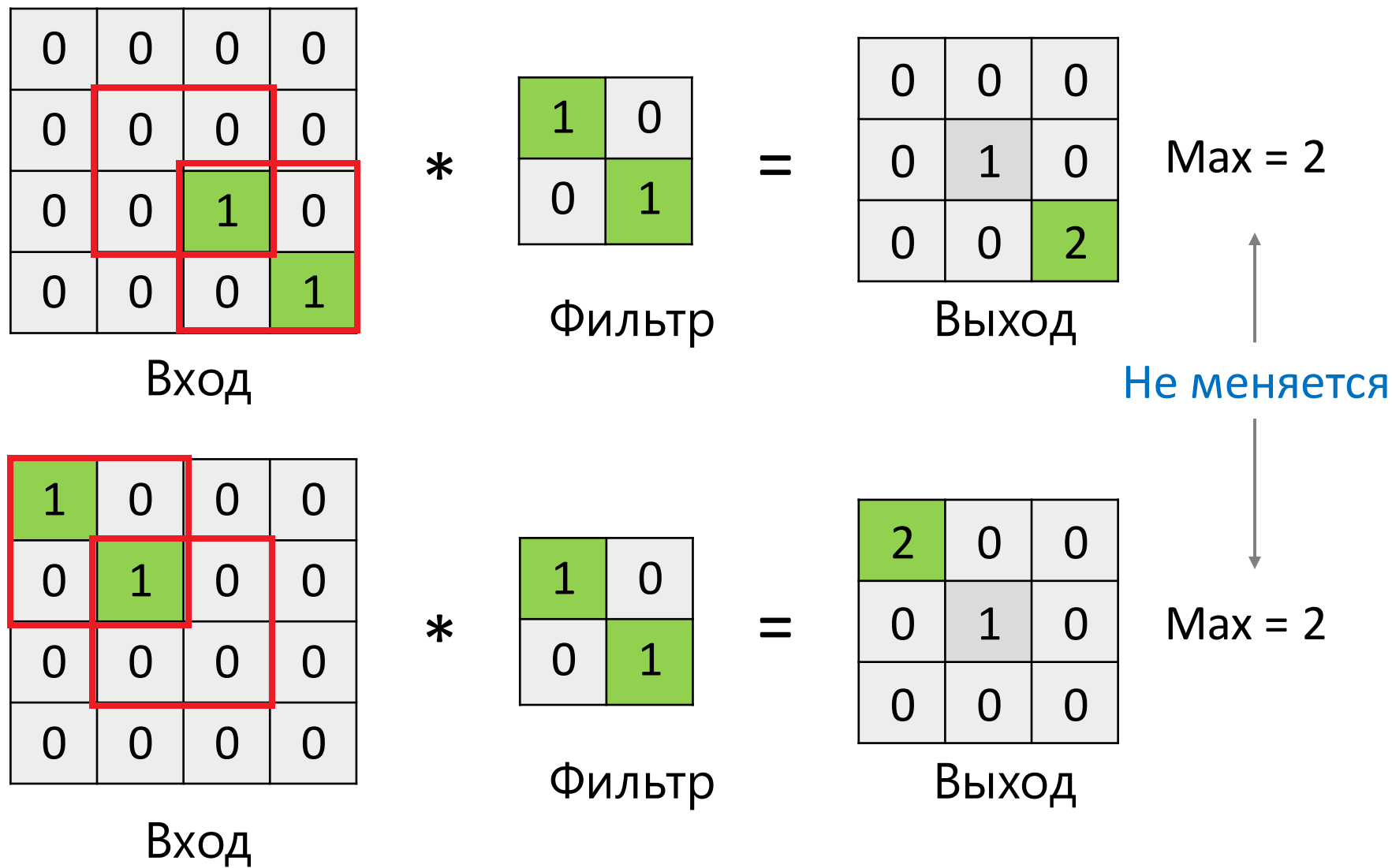
$$\begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \boxed{6}$$

$$\begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \boxed{10}$$

Свёртка

- Операция свёртки выявляет наличие на изображении паттерна, который задаётся фильтром
- Чем сильнее на участке изображения представлен паттерн, тем больше будет значение свёртки

Максимум свёртки инвариантен к сдвигам

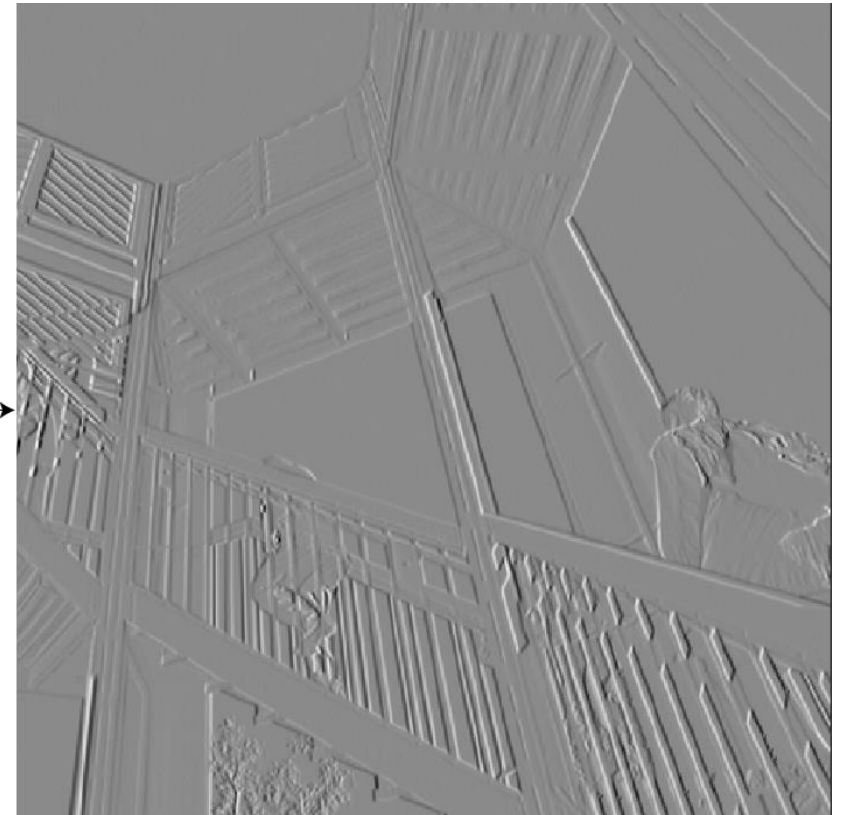


Свёртки в компьютерном зрении



$$\begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix}$$

Horizontal Sobel kernel

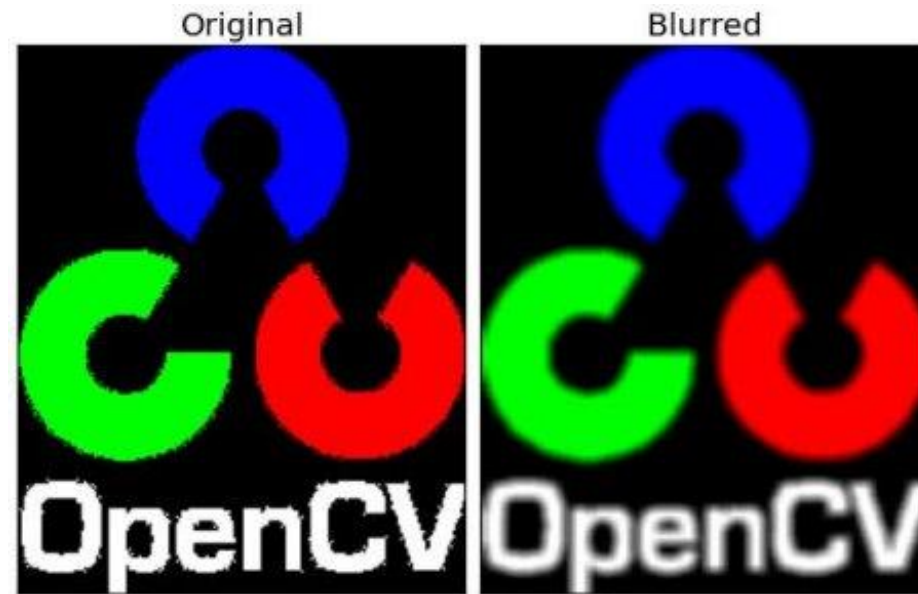


Свёртки в компьютерном зрении



$$\begin{bmatrix} \bullet 0 & \bullet 0 & \bullet 0 \\ \bullet 0 & \bullet 1 & \bullet 0 \\ \bullet 0 & \bullet 0 & \bullet 0 \end{bmatrix} + \begin{bmatrix} \bullet 0 & \bullet 0 & \bullet 0 \\ \bullet 0 & \bullet 1 & \bullet 0 \\ \bullet 0 & \bullet 0 & \bullet 0 \end{bmatrix} - \frac{1}{9} \begin{bmatrix} \bullet 1 & \bullet 1 & \bullet 1 \\ \bullet 1 & \bullet 1 & \bullet 1 \\ \bullet 1 & \bullet 1 & \bullet 1 \end{bmatrix} = \begin{bmatrix} \bullet 0 & \bullet 0 & \bullet 0 \\ \bullet 0 & \bullet 2 & \bullet 0 \\ \bullet 0 & \bullet 0 & \bullet 0 \end{bmatrix} - \frac{1}{9} \begin{bmatrix} \bullet 1 & \bullet 1 & \bullet 1 \\ \bullet 1 & \bullet 1 & \bullet 1 \\ \bullet 1 & \bullet 1 & \bullet 1 \end{bmatrix}$$

Свёртки в компьютерном зрении



$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Свёртка

$$\text{Im}^{out}(x, y) = \sum_{i=-d}^d \sum_{j=-d}^d (K(i, j) \text{Im}^{in}(x + i, y + j) + b)$$

Свёртка

$$\text{Im}^{out}(x, y) = \sum_{i=-d}^d \sum_{j=-d}^d (K(i, j) \text{Im}^{in}(x + i, y + j) + b)$$

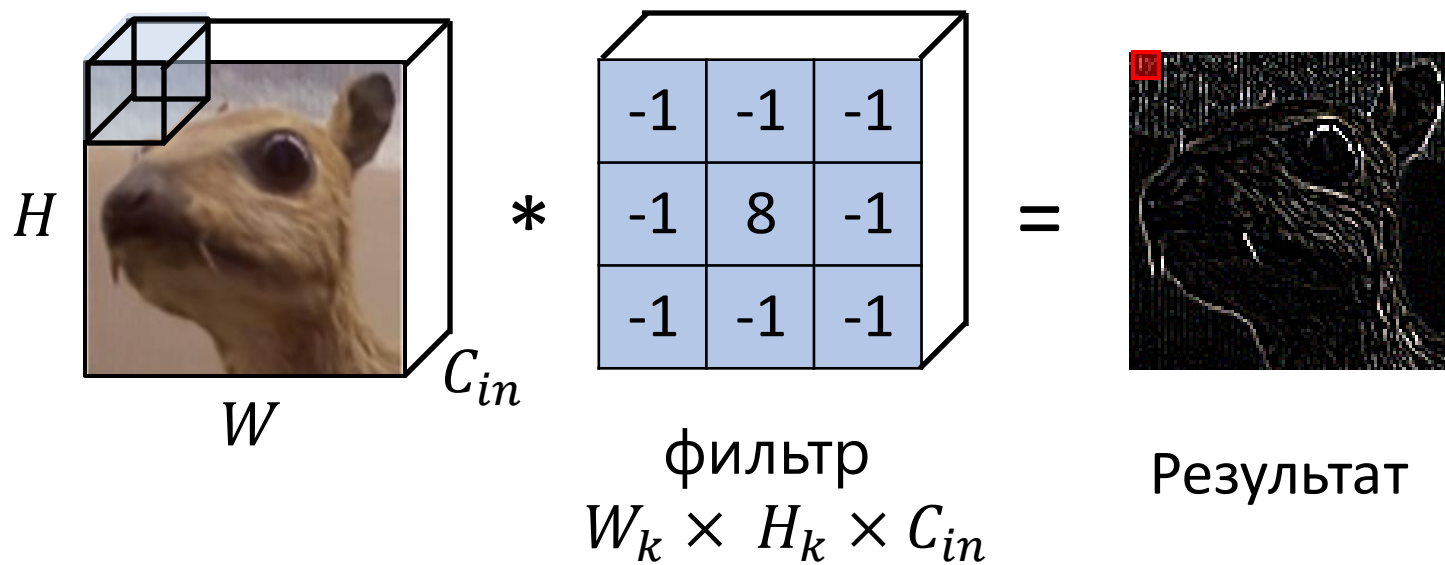
- Пиксель в результирующем изображении зависит только от небольшого участка исходного изображения (local connectivity)
- Веса одни и те же для всех пикселей результирующего изображения (shared weights)

Свёртка

- Обычно исходное изображение цветное!
- Это означает, что в нём несколько каналов (R, G, B)
- Учтём в формуле:

$$\text{Im}^{out}(x, y) = \sum_{i=-d}^d \sum_{j=-d}^d \sum_{c=1}^c (K(i, j, c) \text{Im}^{in}(x + i, y + j, c) + b)$$

Свёртка



Свёртка

- Одна свёртка выделяет конкретный паттерн на изображении
- Нам интересно искать много паттернов
- Сделаем результат трёхмерным:

$$\text{Im}^{out}(x, y, t) = \sum_{i=-d}^d \sum_{j=-d}^d \sum_{c=1}^C (K_t(i, j, c) \text{Im}^{in}(x + i, y + j, c) + b_t)$$

Число параметров

$$\text{Im}^{out}(x, y, t) = \sum_{i=-d}^d \sum_{j=-d}^d \sum_{c=1}^C (\textcolor{red}{K_t(i, j, c)} \text{Im}^{in}(x + i, y + j, c) + \textcolor{red}{b_t})$$

- Обучается только фильтр
- $((2d + 1)^2 * C + 1) * T$ параметров