

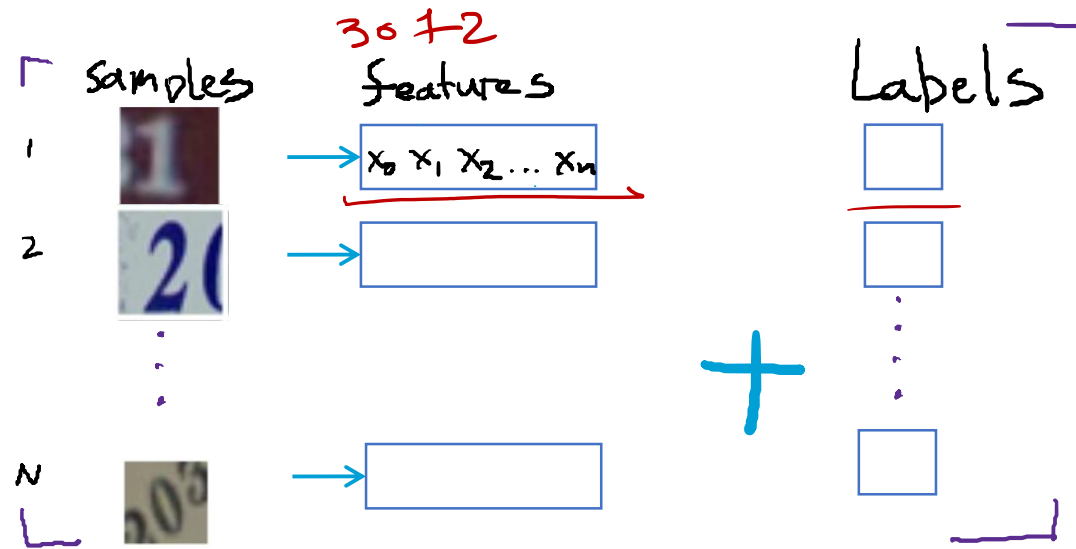
Лекция 3

Нейронные Сети

В предыдущих сериях

training data

predict

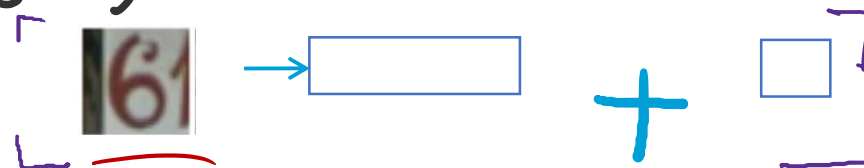


ML
algorithm

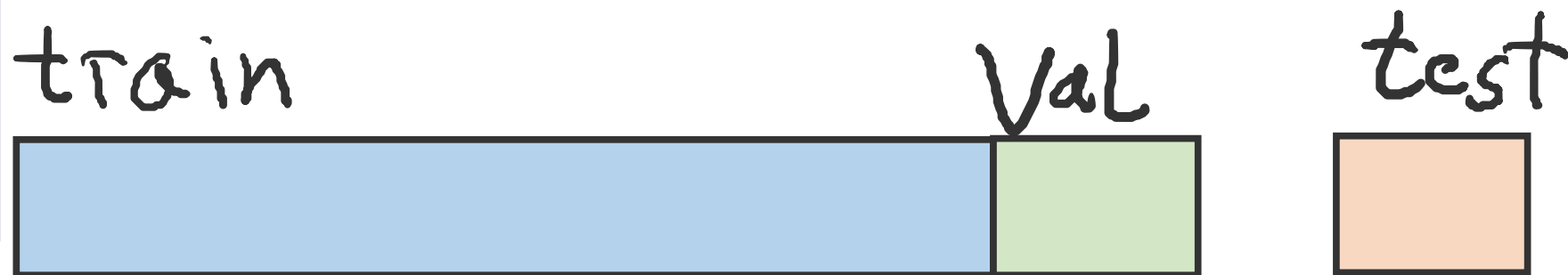


Model

test



Machine Learning Flow



underfitting

- Более мощную модель
- Больше ресурсов для тренировки
- Другой подход

overfitting

- Больше данных
- Больше регуляризации
- Другой подход

- Отличаются train и test
- Больше данных, таких как test

Ошибка на train

большая

маленькая

Ошибка на val

большая

маленькая

Ошибка на test

большая

маленькая

Кстати вот он про это подробнее:

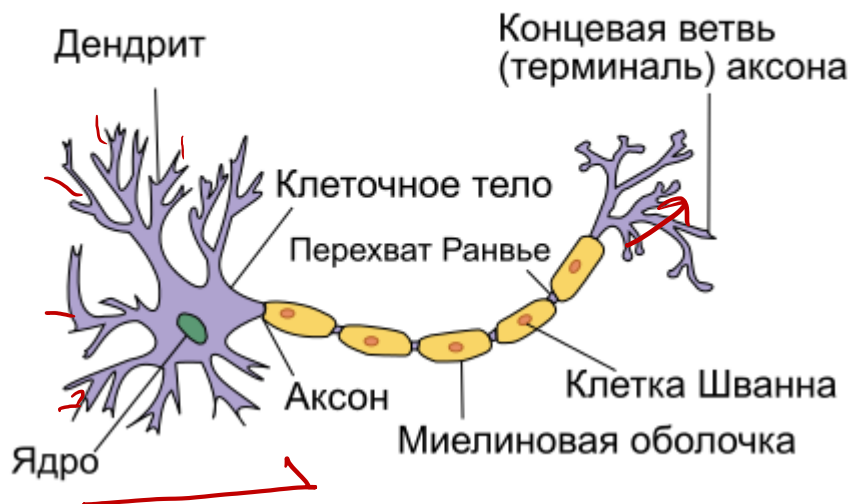
[Nuts and Bolts of Applying Deep Learning](#)



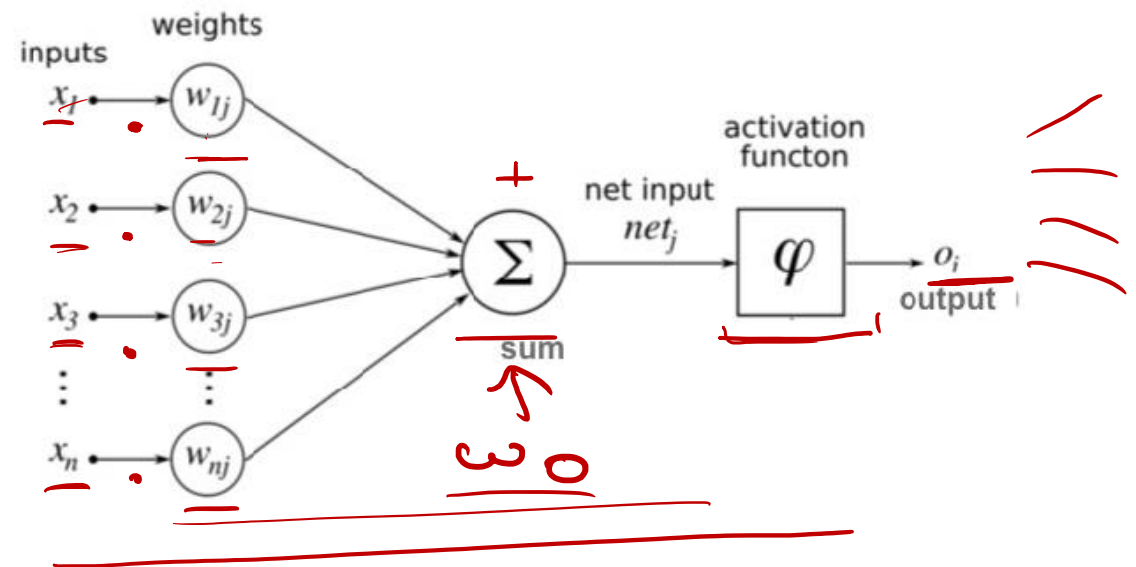
Нейронная сеть

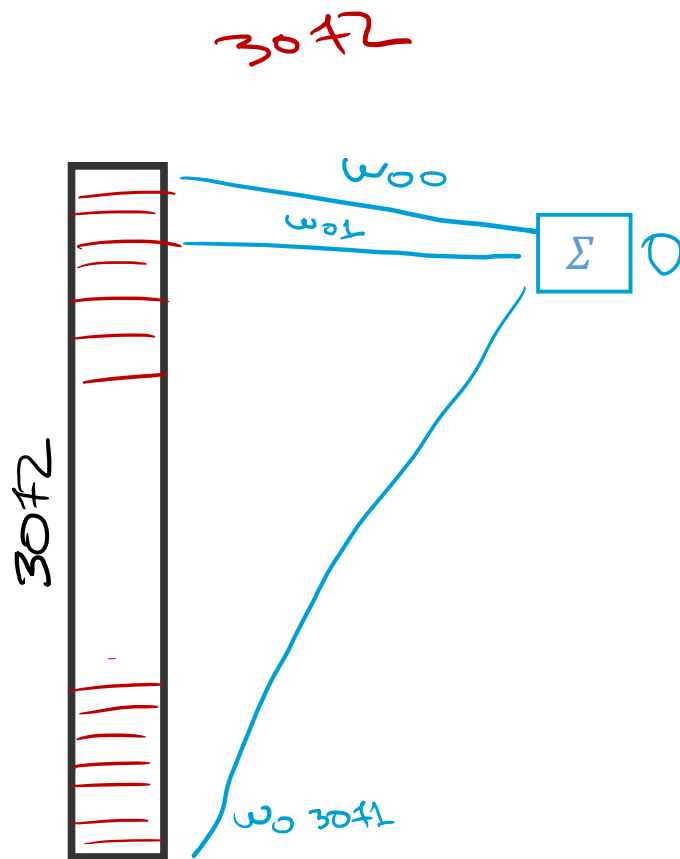
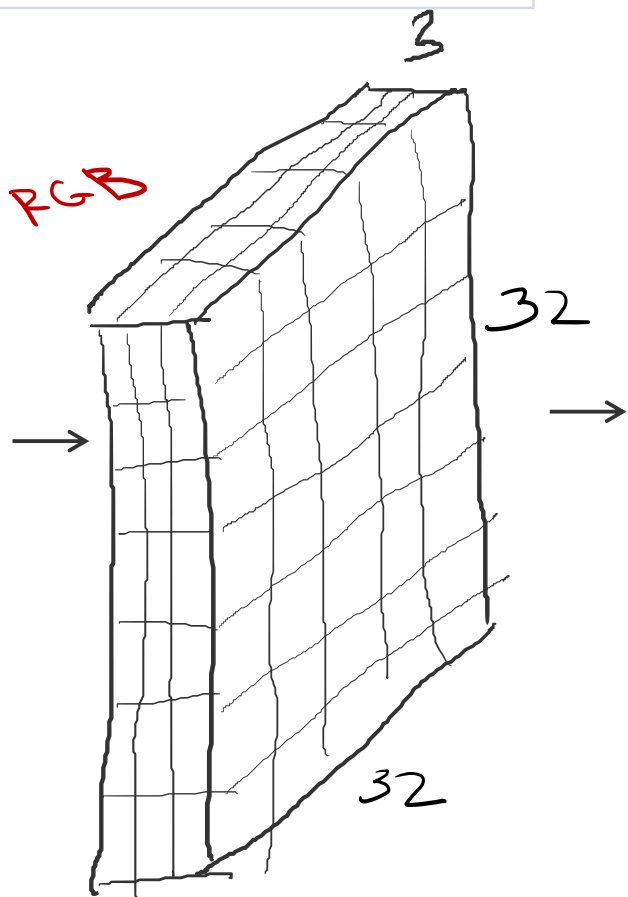
Neural network

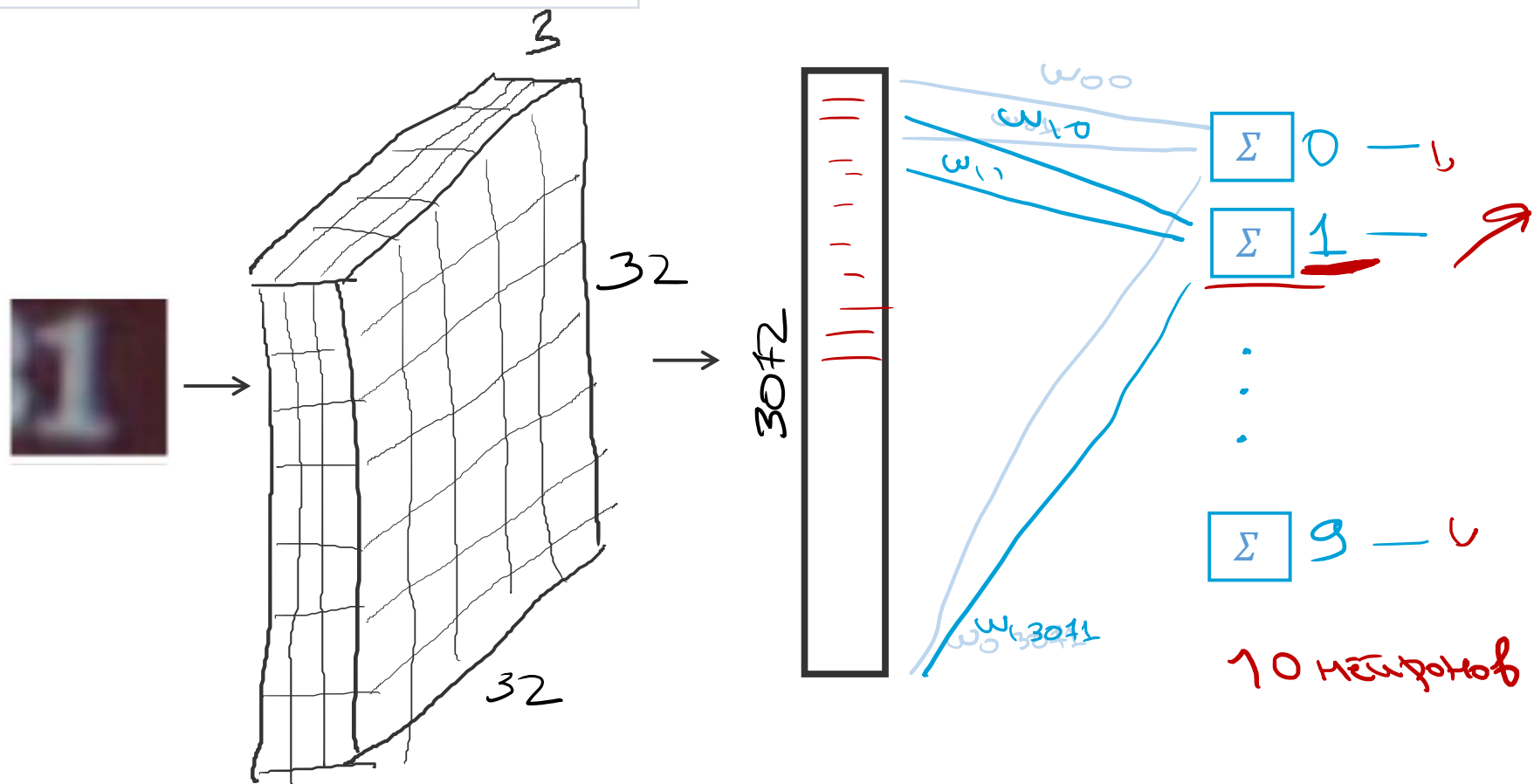
Типичная структура нейрона

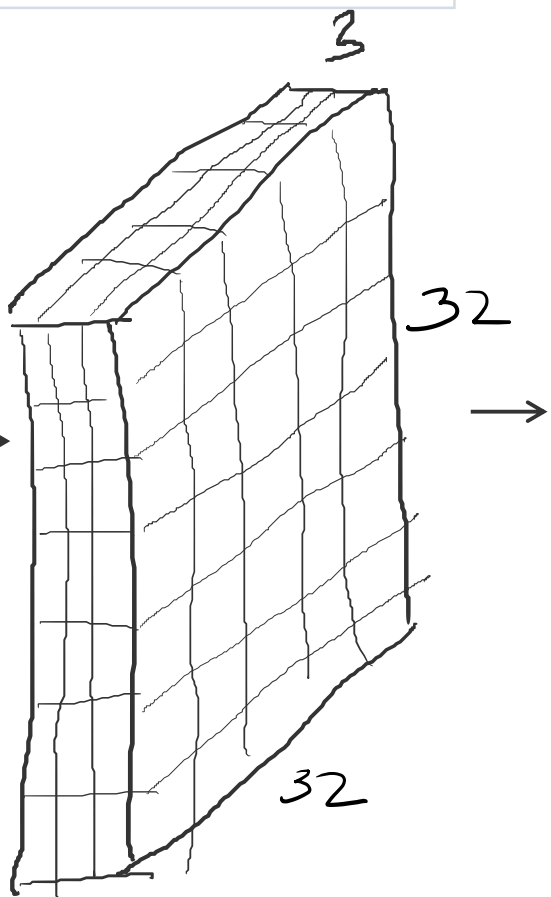


[Wikipedia](https://ru.wikipedia.org/wiki/Нейрон)

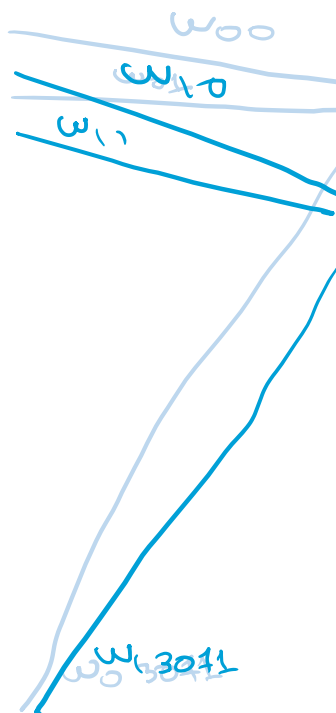








3072



$$\Sigma 0 -$$

$$\Sigma 1 -$$

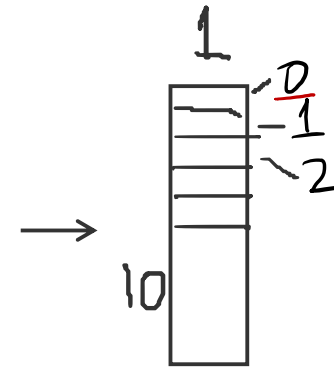
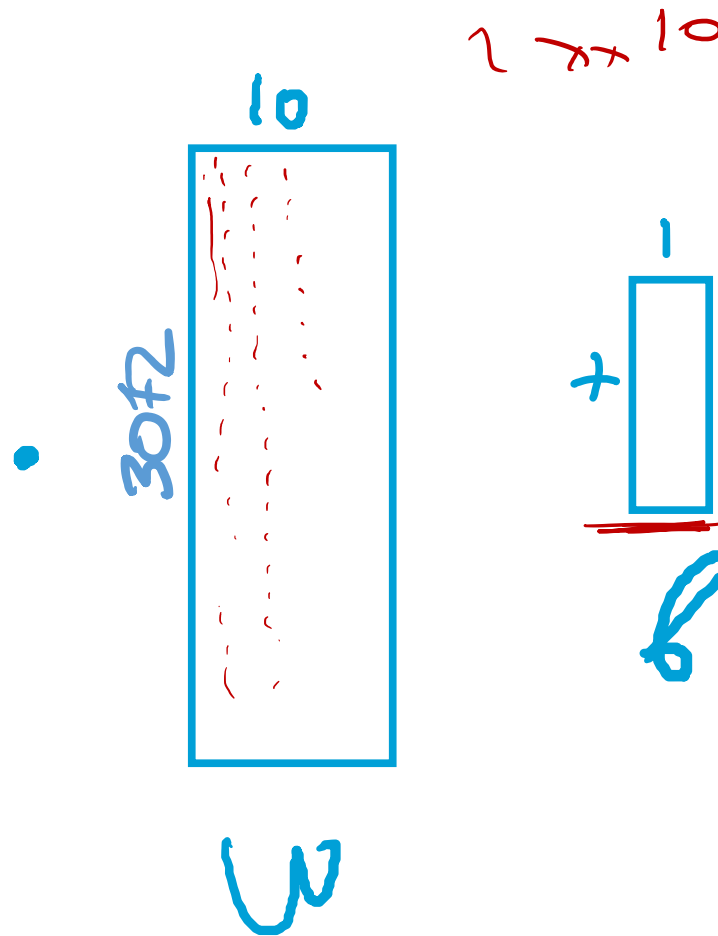
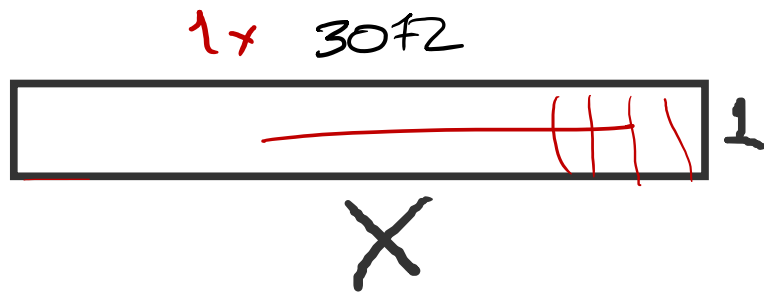
...

$$\Sigma 9 -$$

+C



Линейный классификатор Linear classifier



$$\underline{y = x \cdot w + b}$$

Линейный классификатор Linear classifier

$$0.4 \cdot 0.5 + 0.2 \cdot 1 + 0 \cdot 0 + -1 \cdot 0 + 1$$

$$\begin{matrix} & 4 \\ \begin{bmatrix} \underline{0.5} & \underline{1} & \underline{0} & \underline{0} \end{bmatrix} & \cdot \end{matrix}$$



$$\cdot \begin{matrix} & 2 \\ \begin{bmatrix} \underline{1} & \underline{0.4} \\ \underline{-1} & \underline{0.2} \\ \underline{0.3} & \underline{0} \\ \underline{2} & \underline{-1} \end{bmatrix} \end{matrix}$$

w

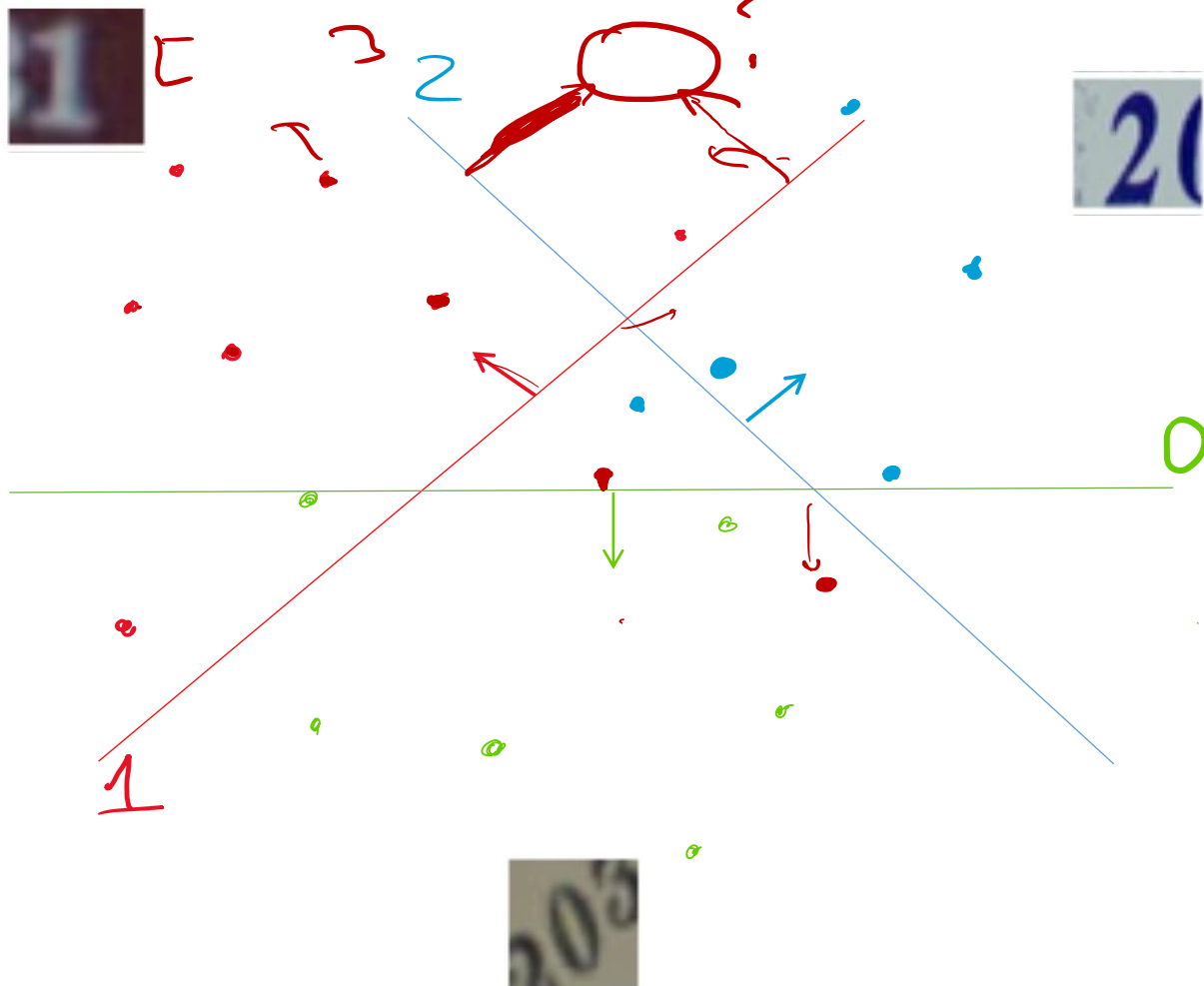
$$+ \begin{matrix} & 1 \\ \begin{bmatrix} \underline{0} \\ \underline{1} \end{bmatrix} \end{matrix} \cdot 2 = \begin{matrix} & 2 \\ \begin{bmatrix} \underline{-0.5} \\ \underline{1.4} \end{bmatrix} \end{matrix}$$

multi-class

Разделяющие плоскости

3042

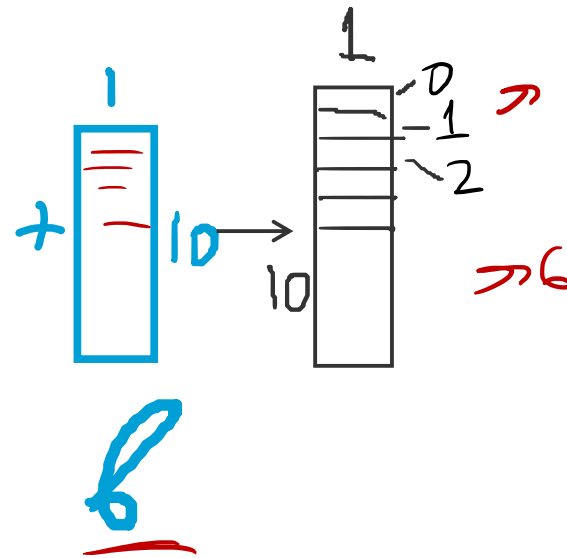
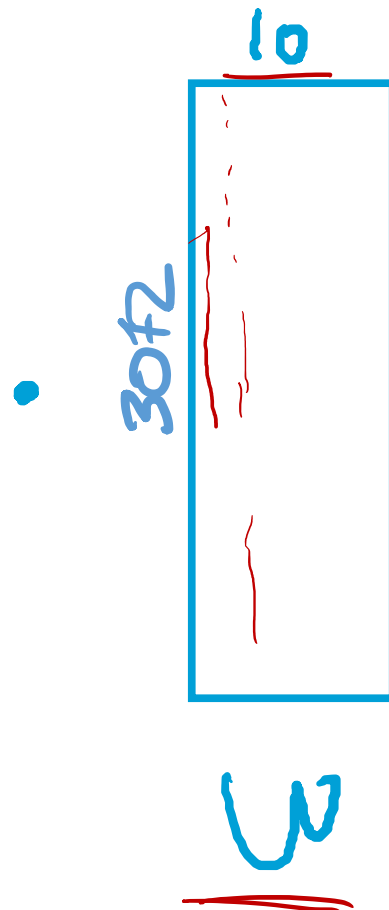
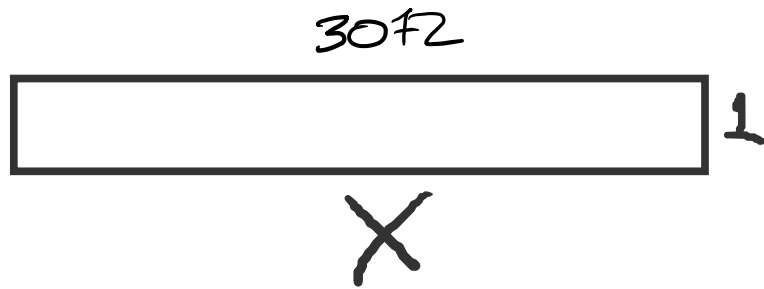
3042 - мерная



ω, b

Найти лучшие w и b

$$\underline{3042 \cdot 10}$$

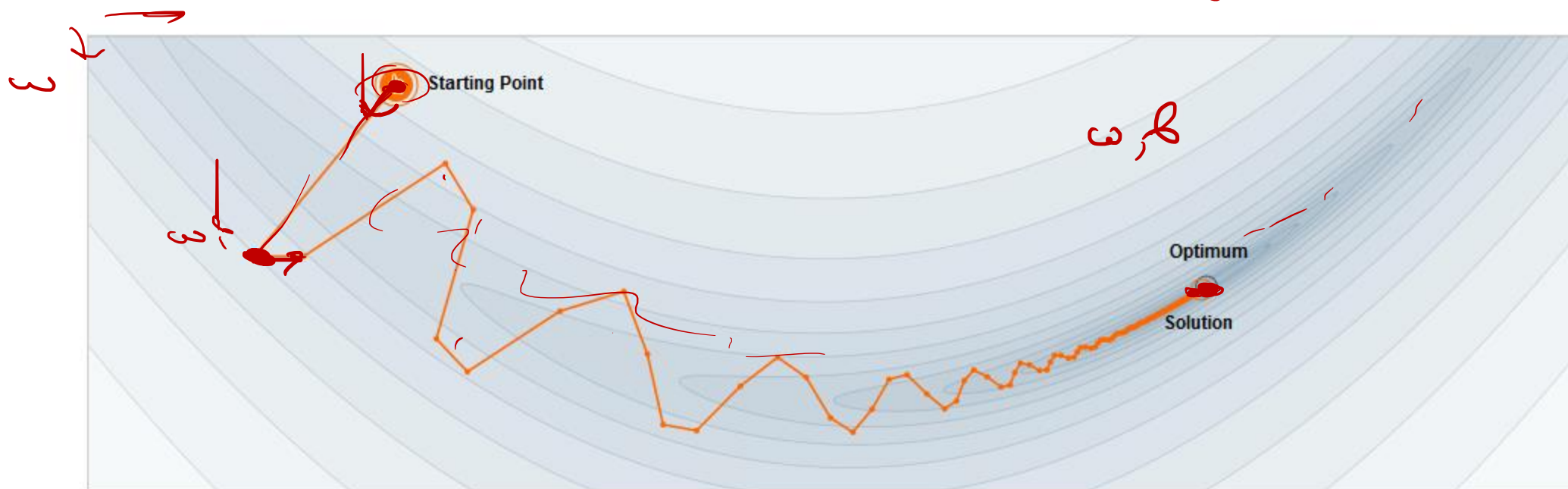


Градиентный спуск

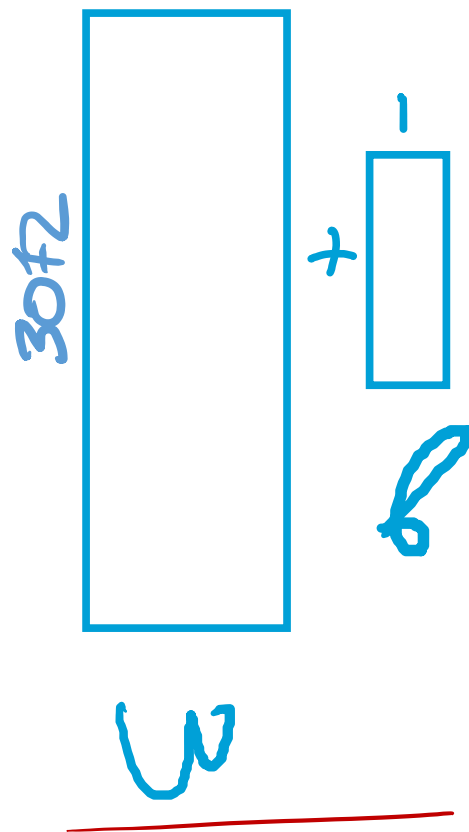
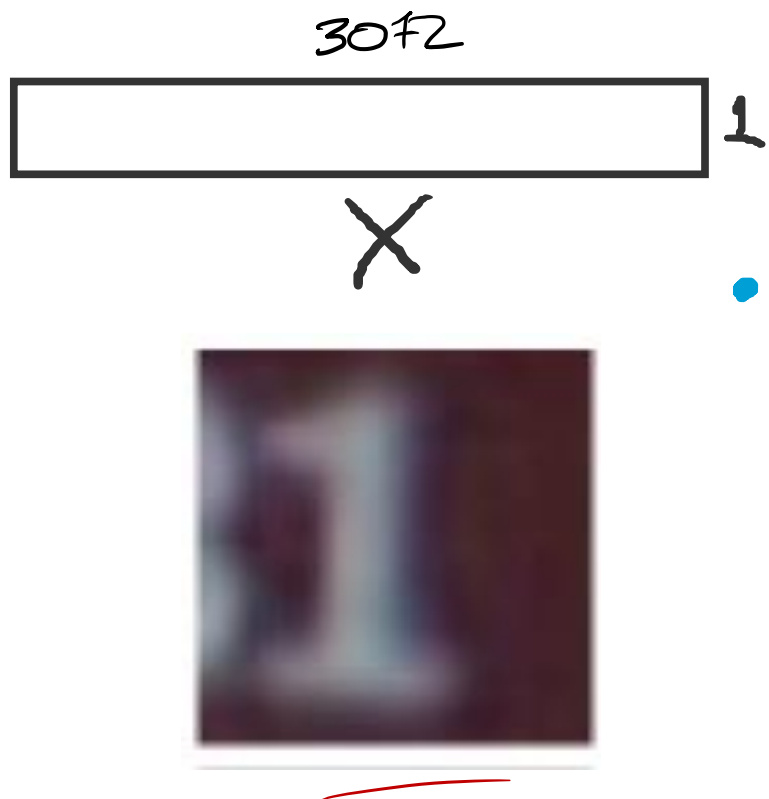
Gradient descent



$$\frac{\omega, b}{3042 \cdot 10 + 3042}$$
$$\frac{\text{grad} \cdot \lambda}{b}$$



Softmax



Y

0.5
-5.0
0
1.3

$y = x \cdot w + b$

Softmax

$p(C=0|x)$

0.3
0
0.1
0.5

0.99

0.1

$p(C=0|x) = \frac{e^{y_0}}{e^{y_0} + e^{y_1} + \dots + e^{y_n}} = \frac{e^{y_0}}{\sum_i e^{y_i}}$

$p(C=1|x) = \frac{e^{y_1}}{e^{y_0} + e^{y_1} + \dots + e^{y_n}} = \frac{e^{y_1}}{\sum_i e^{y_i}}$

+0: ~~0~~

0-1

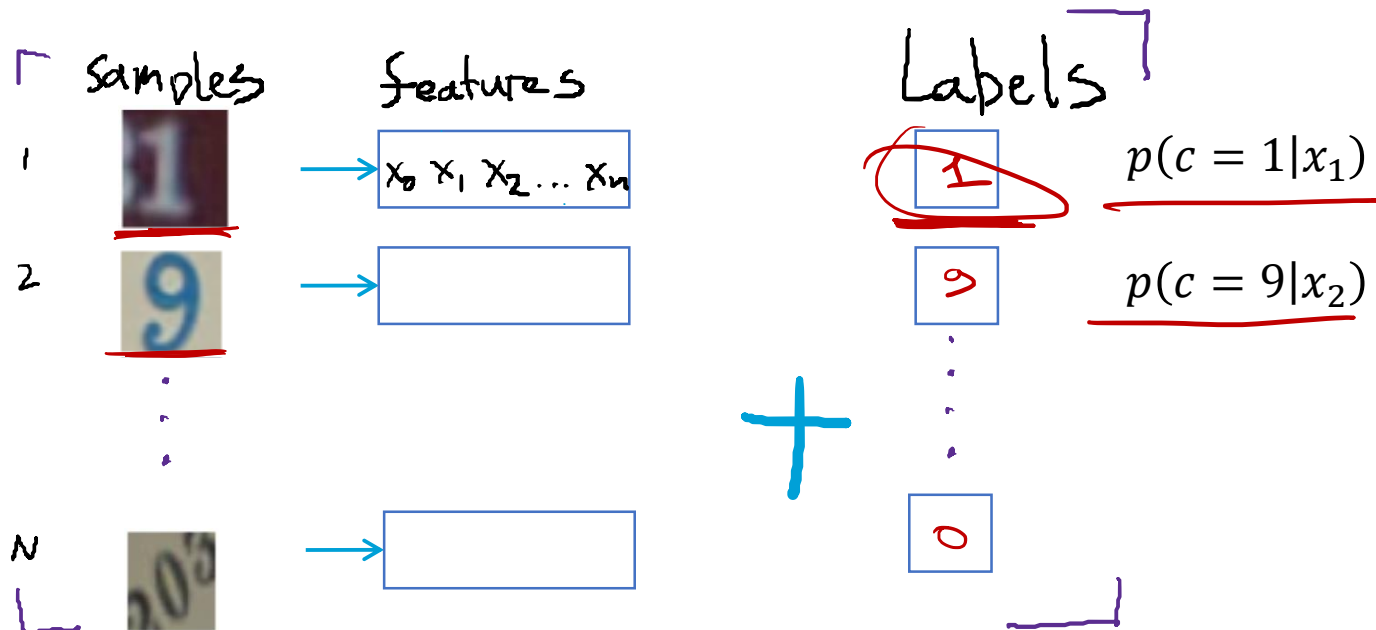
-0.00

Принцип максимального правдоподобия

Maximum likelihood

ground truth

training data



$$p(data) = \prod_s p(c = gt_s | x_s) \quad \xrightarrow{w, b}$$

Negative Log-likelihood:

$$-\ln p(data) = - \sum_s \ln p(c = gt_s | x_s) \quad \xrightarrow{w, b}$$

no w, b

Cross-Entropy loss

$$= - \sum_s \ln \frac{e^{(wx_s+b)_{gt_s}}}{\sum_i e^{(wx_s+b)_i}} = L \text{ loss}$$

$s \rightarrow \text{sample}$

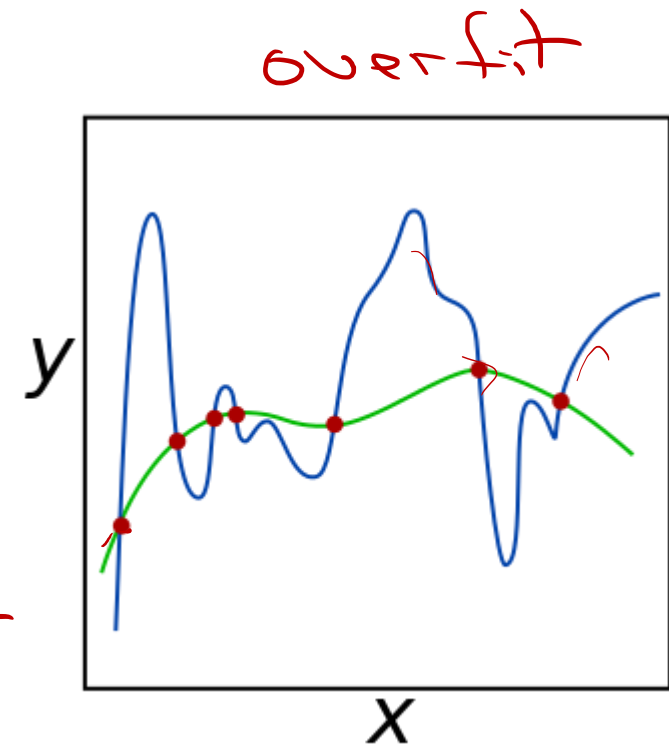
Регуляризация Regularization

$$\underline{L = - \sum_s \ln \frac{e^{(wx_s+b)gt_s}}{\sum_i e^{(wx_s+b)_i}} + \lambda R(w, b)}$$

$b \neq 2$

$$R(w, b) = \underline{\underline{\|w\|_2^2 + \|b\|_2^2}}$$

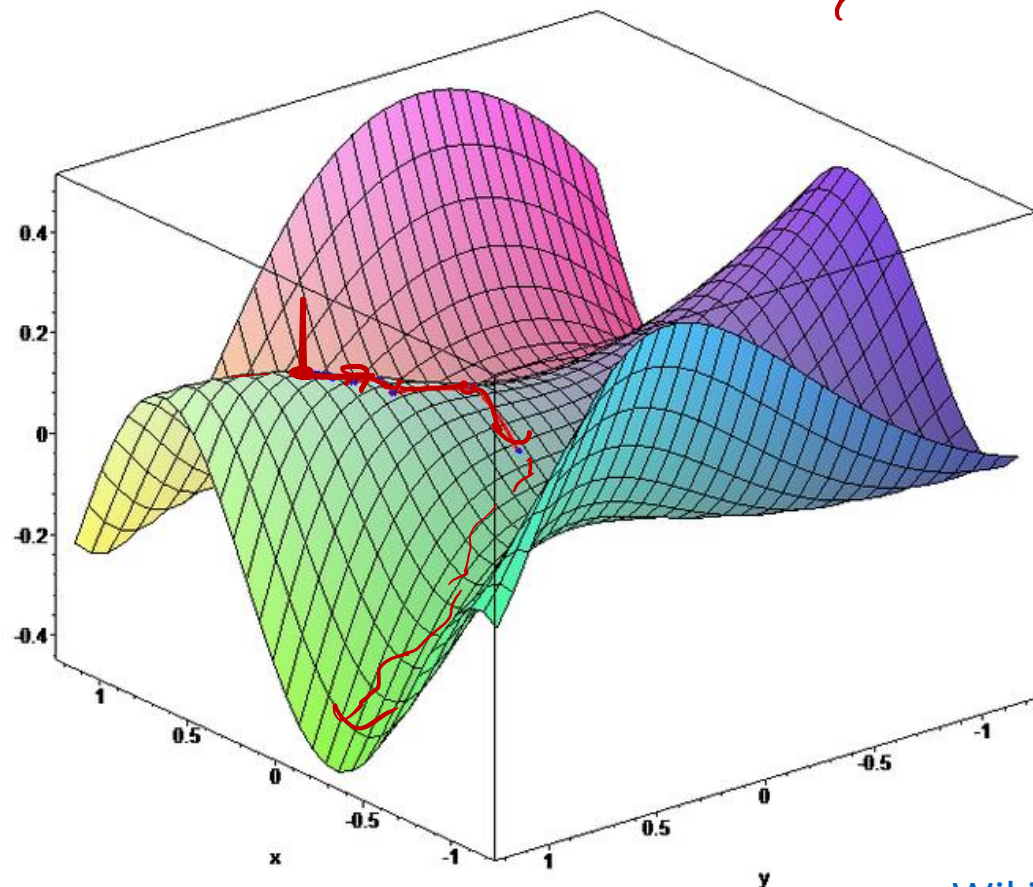
$$\|w\|_2^2 = \omega_{00}^2 + \omega_{01}^2 + \dots$$



[Wikipedia](#)

Gradient descent

Градиентный спуск



$$\underline{\vec{w}} = \underline{\vec{w}} - \eta \underline{\vec{\nabla}_w L}$$

$$\underline{\vec{b}} = \underline{\vec{b}} - \eta \underline{\vec{\nabla}_b L}$$

finite differences

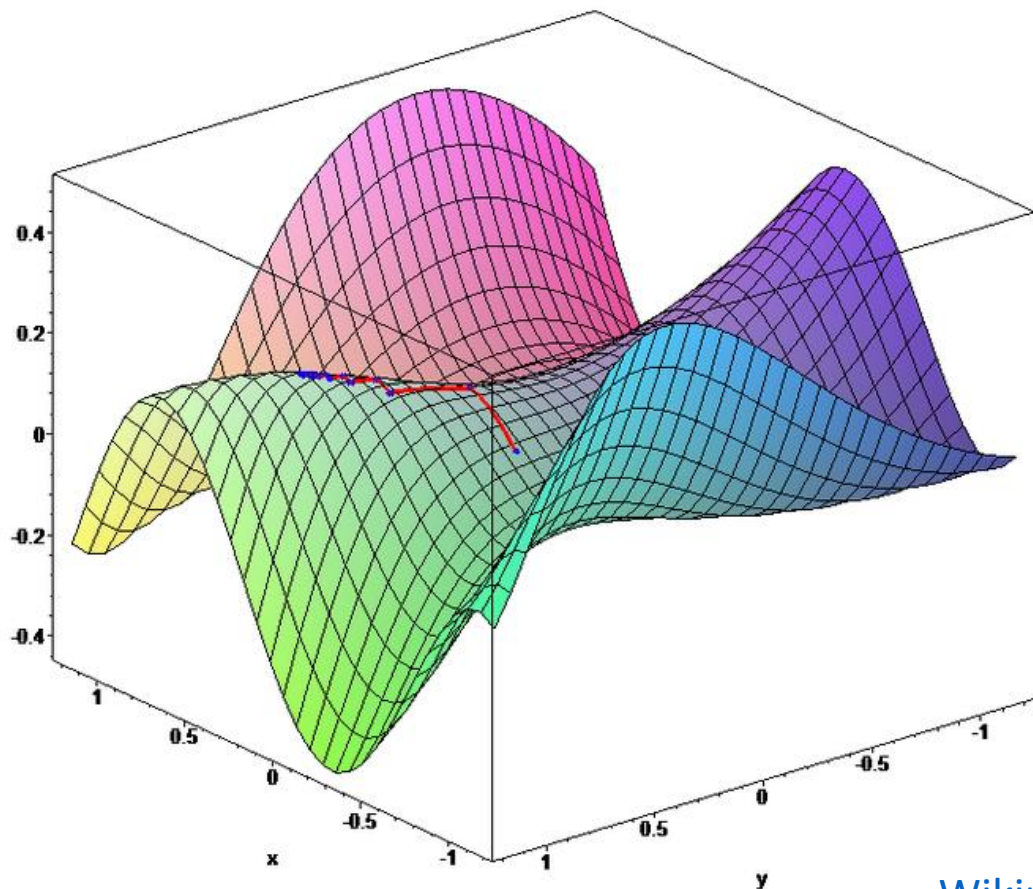
$$\frac{L(x + \varepsilon) - L(x - \varepsilon)}{2\varepsilon}$$

Handwritten notes: w, b (above the formula), $10 + \varepsilon$ (above the first x), 10 (below the first x), 10 (below the second x), 10 (below the denominator 2ε).

$$L'(x) \approx \frac{L(x + \varepsilon) - L(x - \varepsilon)}{2\varepsilon}$$

Gradient descent

Градиентный спуск



$$\vec{w} = \vec{w} - \eta \vec{\nabla}_w L$$

$$\vec{b} = \vec{b} - \eta \vec{\nabla}_b L$$

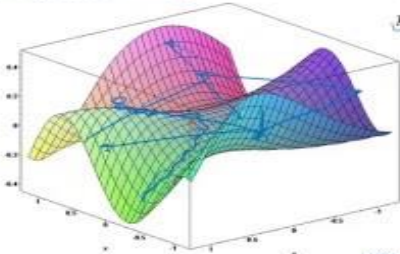
Analytic gradient

$$\frac{\partial}{\partial b_k} \left(- \sum_s \ln \frac{e^{(w x_s + b) g_{ts}}}{\sum_i e^{(w x_s + b)_i}} + \lambda (\|w\|_2^2 + \|b\|_2^2) \right) =$$

Помощь в выводе

Соединяем все вместе

w, b



$$L = -\ln p(\text{data}) = -\sum \ln p(c = y_i | x_i)$$
$$p(c = i | x) = \text{softmax}(\vec{x} \cdot \vec{w} + b) \quad \text{softmax}(\vec{v}) = \frac{e^{v_i}}{\sum_k e^{v_k}}$$

Analytic gradient

$$\nabla_{\vec{w}} \left(-\sum_i \ln \frac{e^{\vec{w}_i \cdot \vec{x}_i + b_i}}{\sum_k e^{\vec{w}_k \cdot \vec{x}_i + b_k}} + \lambda (\|\vec{w}\|_2^2 + \|\vec{b}\|_2^2) \right)$$
$$\vec{w} = \vec{w} - \eta \vec{\nabla}_{\vec{w}} L$$
$$\vec{b} = \vec{b} - \eta \vec{\nabla}_{\vec{b}} L$$

Wikipedia

[Link](#)

<http://cs231n.github.io/linear-classify/>
<http://cs231n.github.io/optimization-1/>

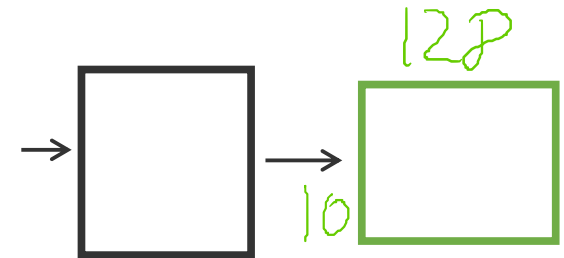
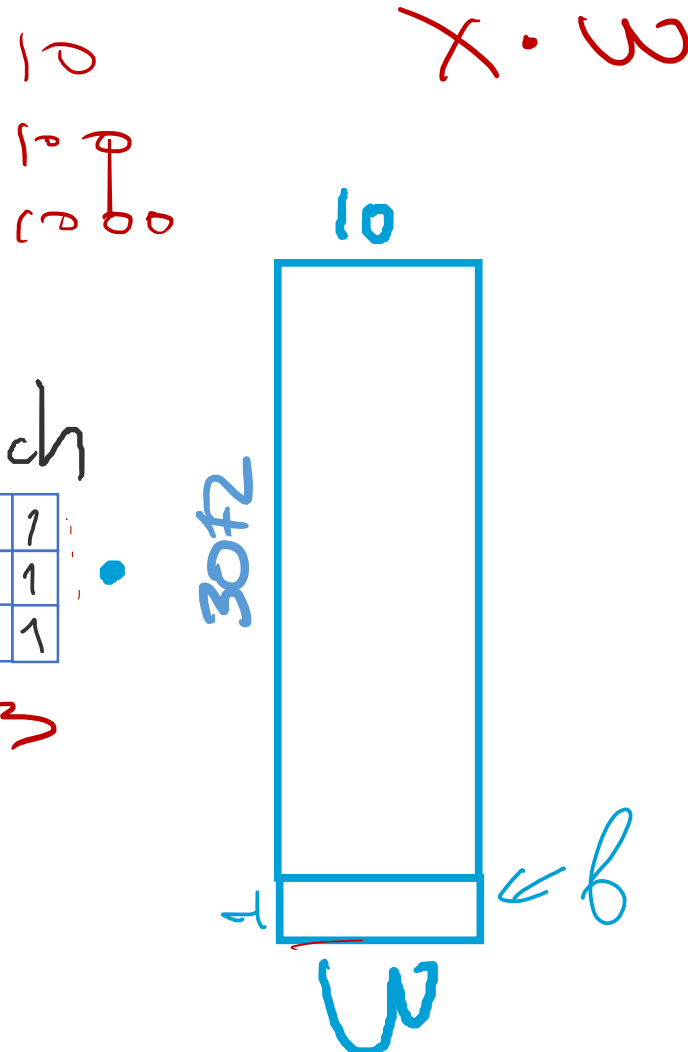
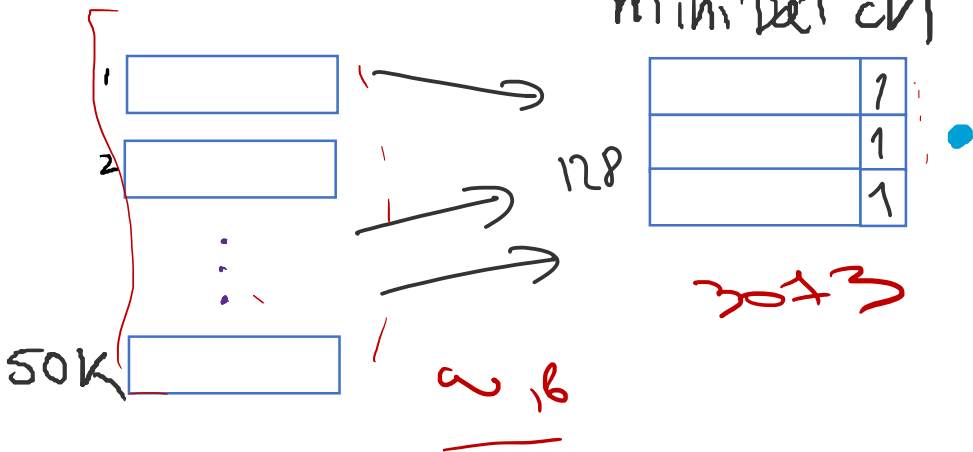
методика

ПОСЛЕДНЮЮ

И ТОЧНО ВСЁ

Стохастический градиентный спуск Stochastic Gradient Descent (SGD)

$$L = - \sum_j \ln p(c = y_j | x_j)$$

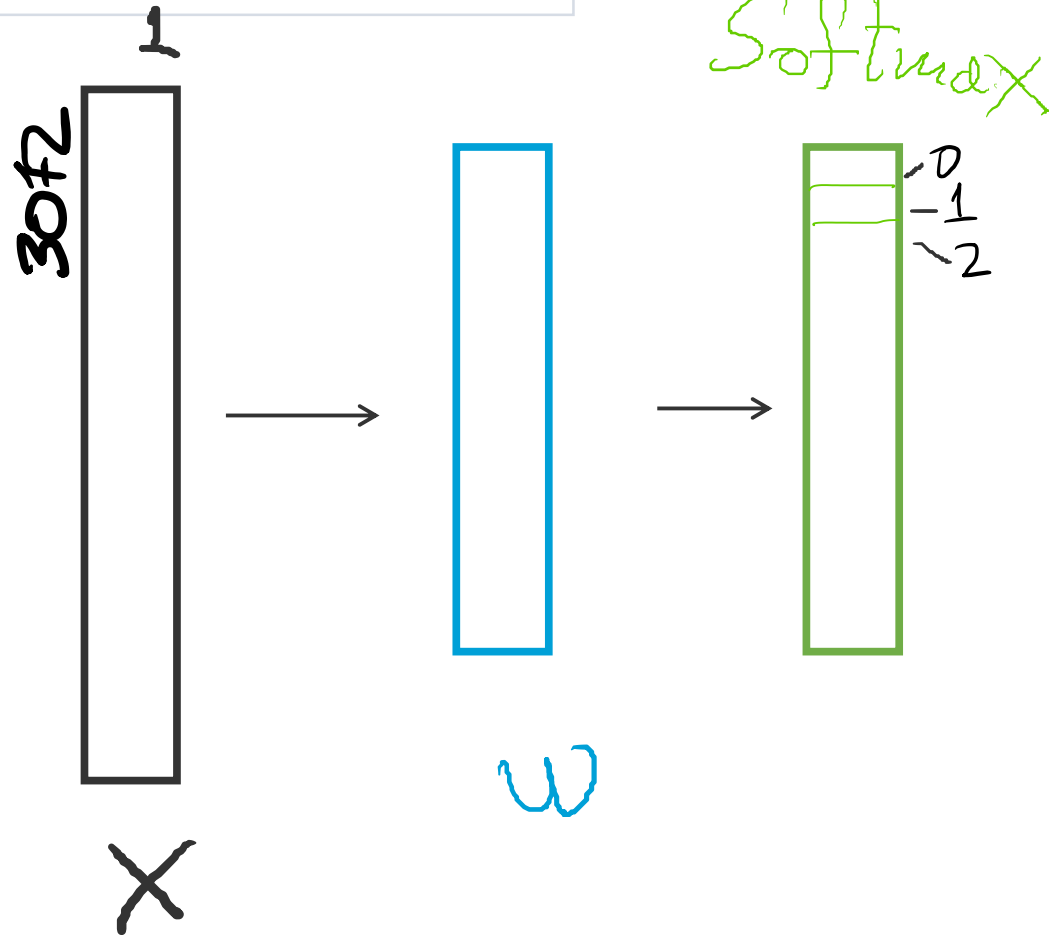



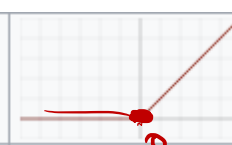
$$\vec{w} = \vec{w} - \eta \nabla_{\vec{w}} L$$

ДАЖЕ ЯДЕРНЫЙ РЕАКТОР

**ДЕЛАЕТ ПЕРЕРЫВ В РАБОТЕ.
ОТДОХНИ!**

Так вот, нейронные сети



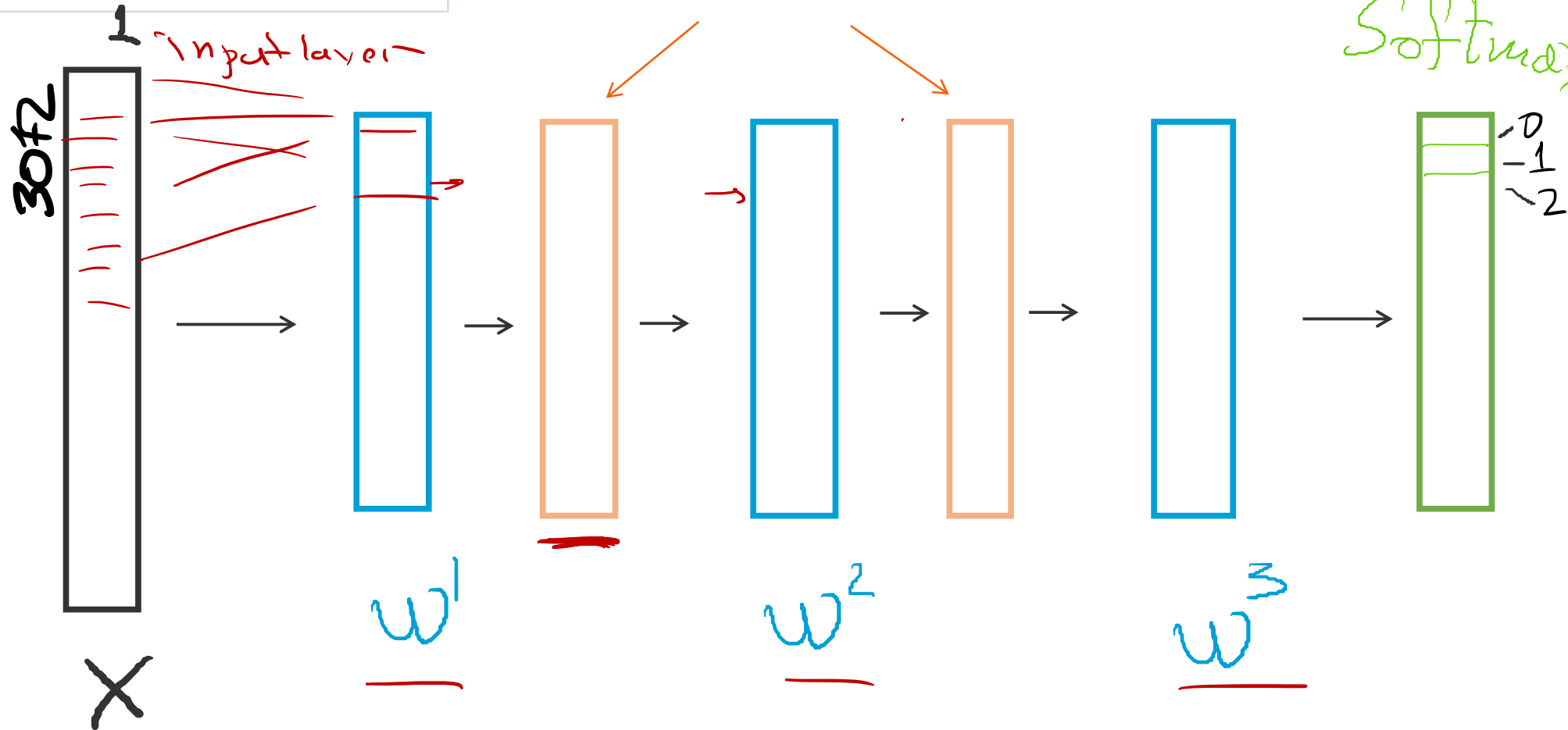
TanH		$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$
Rectified linear unit (ReLU) ^[10]		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$



[Wikipedia](#)

non-linear
function

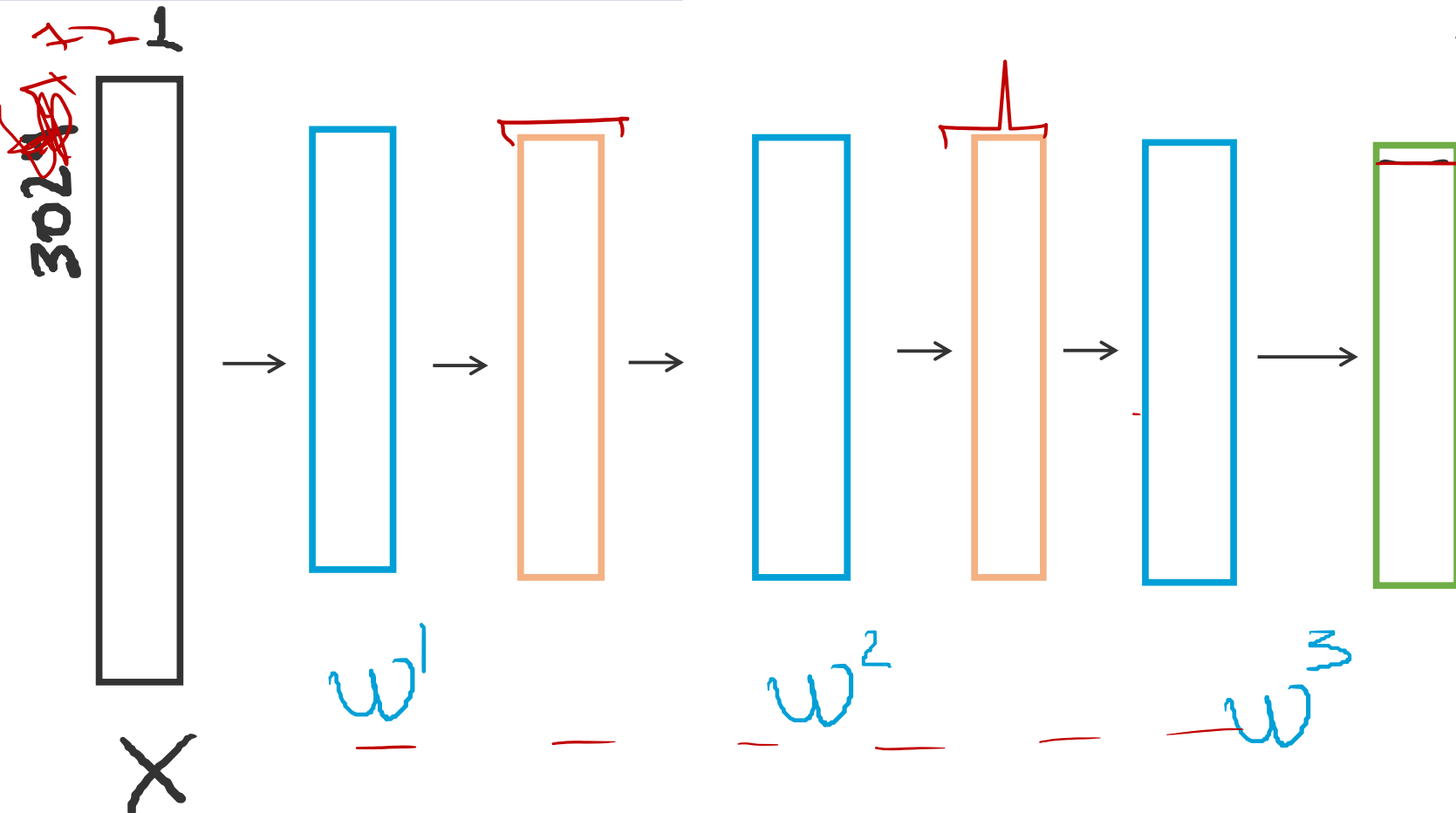
Softmax



Тренировка

100

1000



$$p(c=0|x)$$

$$\underline{L} = - \sum_j \ln p(c = y_j | x_j) + \underline{\lambda R(\omega)}$$

$$\underline{\vec{w}^1} = \underline{\vec{w}^1} - \eta \underline{\vec{\nabla}_{w^1} L}$$

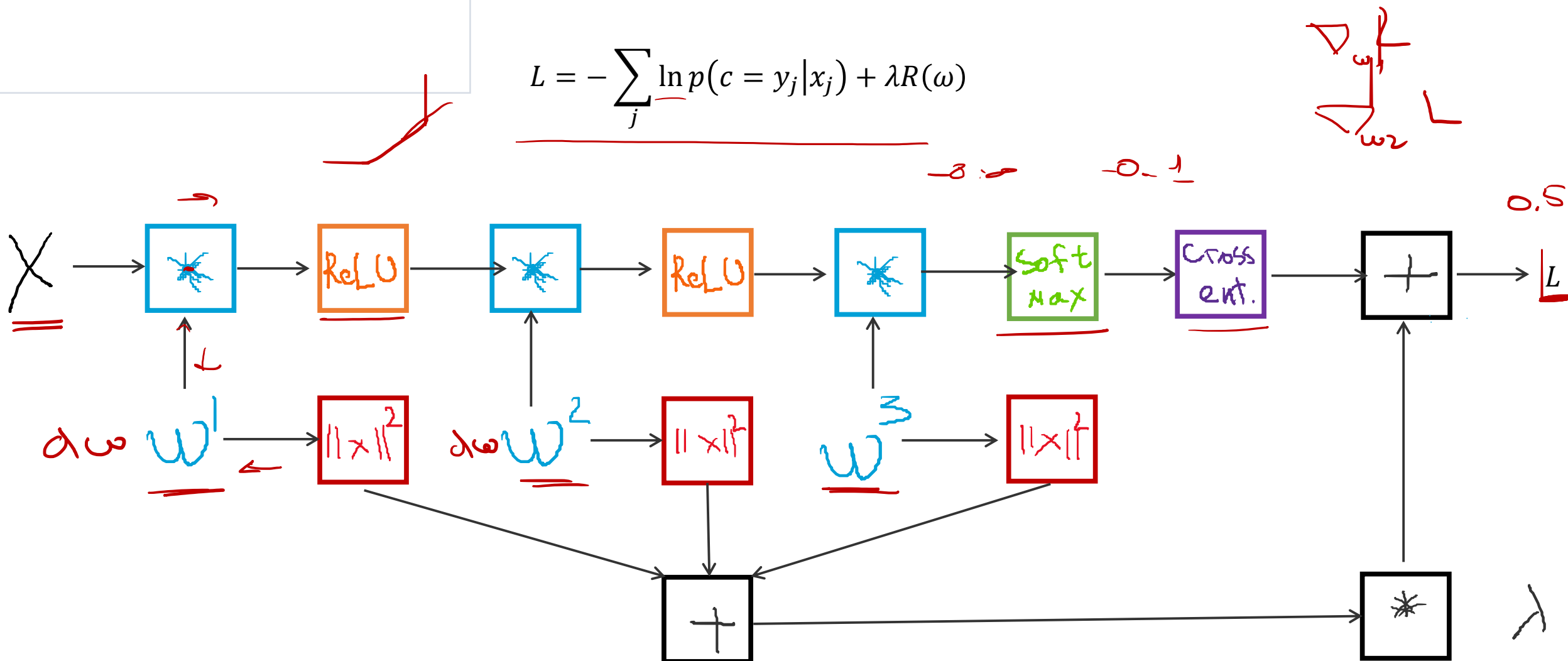
$$\underline{\vec{w}^2} = \underline{\vec{w}^2} - \eta \underline{\vec{\nabla}_{w^2} L}$$

$$\underline{\vec{w}^3} = \underline{\vec{w}^3} - \eta \underline{\vec{\nabla}_{w^3} L}$$

Как посчитать $\vec{\nabla}_w L$?

Граф вычислений Computational graph

$$L = - \sum_j \ln p(c = y_j | x_j) + \lambda R(\omega)$$



Тренируемся на кошках

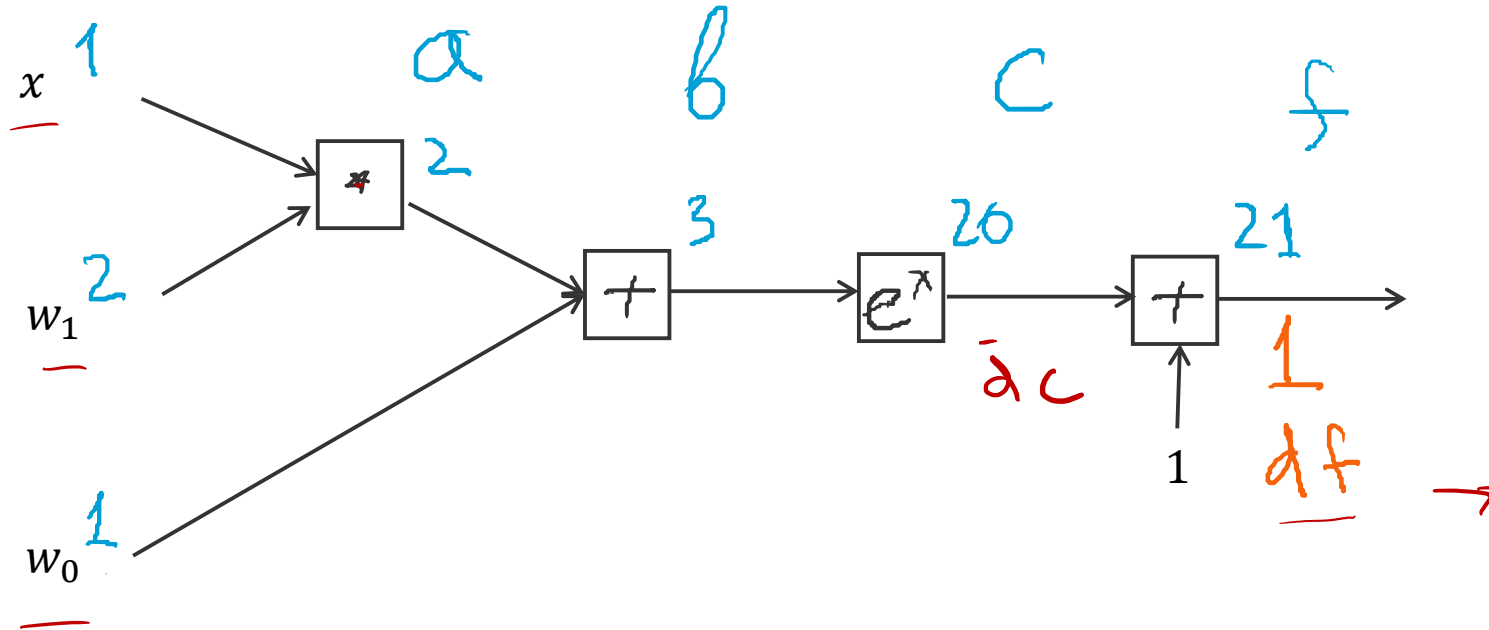
$$f(g(x))$$

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

$$f(x, w) = 1 + e^{w_1 x + w_0}$$

$$f = c + 1$$

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



$$\frac{df}{dx}$$

$$\frac{df}{dw_1}$$

$$\frac{df}{dw_0}$$

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

$$\frac{df}{dw_1}$$

Тренируемся на кошках

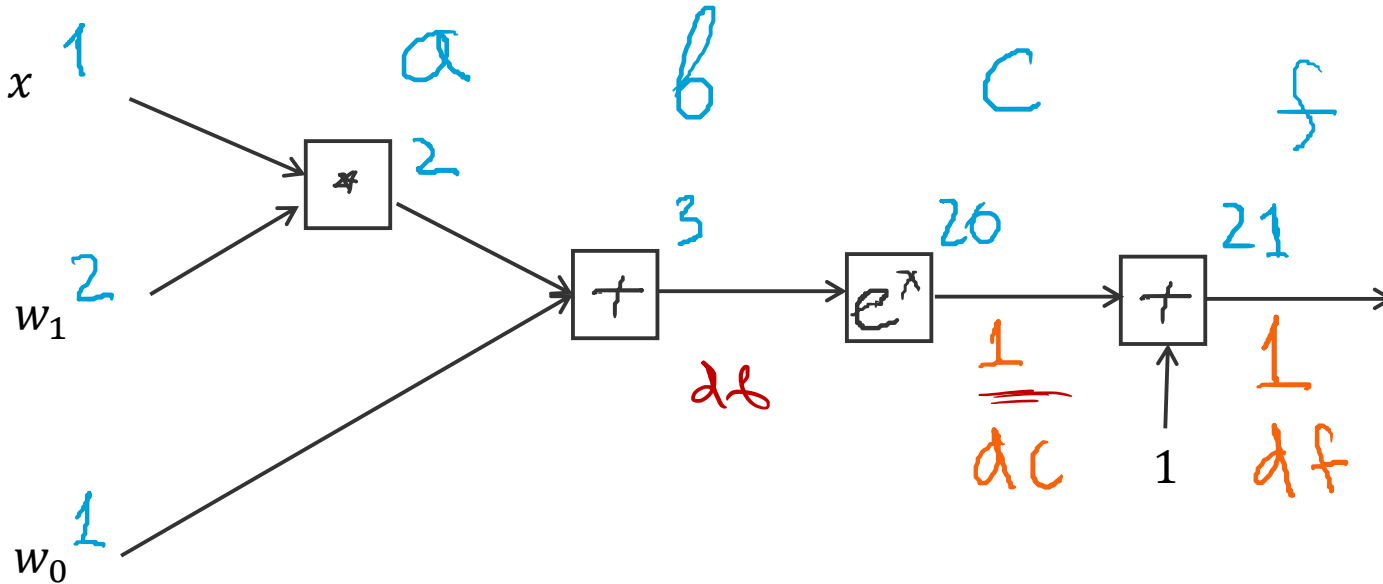
$$f(g(x)) \quad \frac{df}{dx} = \frac{df}{dg} \frac{dg}{dx}$$

$$f(x, w) = 1 + e^{w_1 x + w_0}$$

$$\frac{df}{db} = \frac{df}{dc} \cdot \frac{dc}{db}$$

$\frac{df}{dc} = 1$
 $\frac{dc}{db} = e^b$

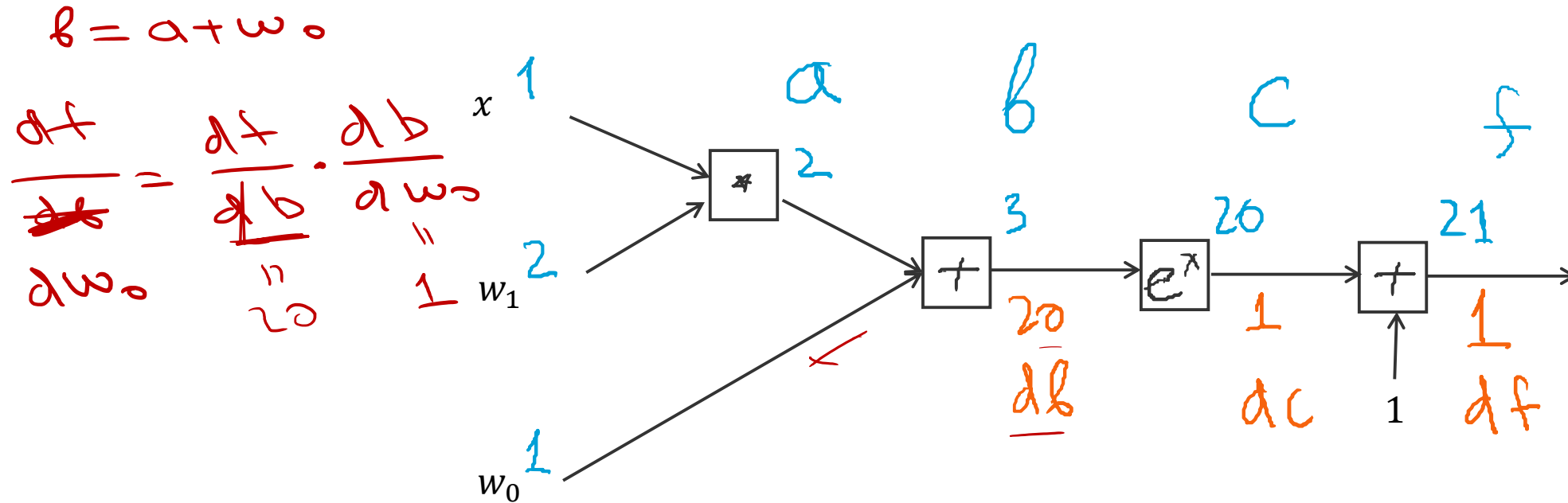
$$c = e^b$$



Тренируемся на кошках

$$f(g(x)) \quad \frac{df}{dx} = \frac{df}{dg} \frac{dg}{dx}$$

$$f(x, w) = 1 + e^{w_1 x + w_0}$$



Тренируемся на кошках

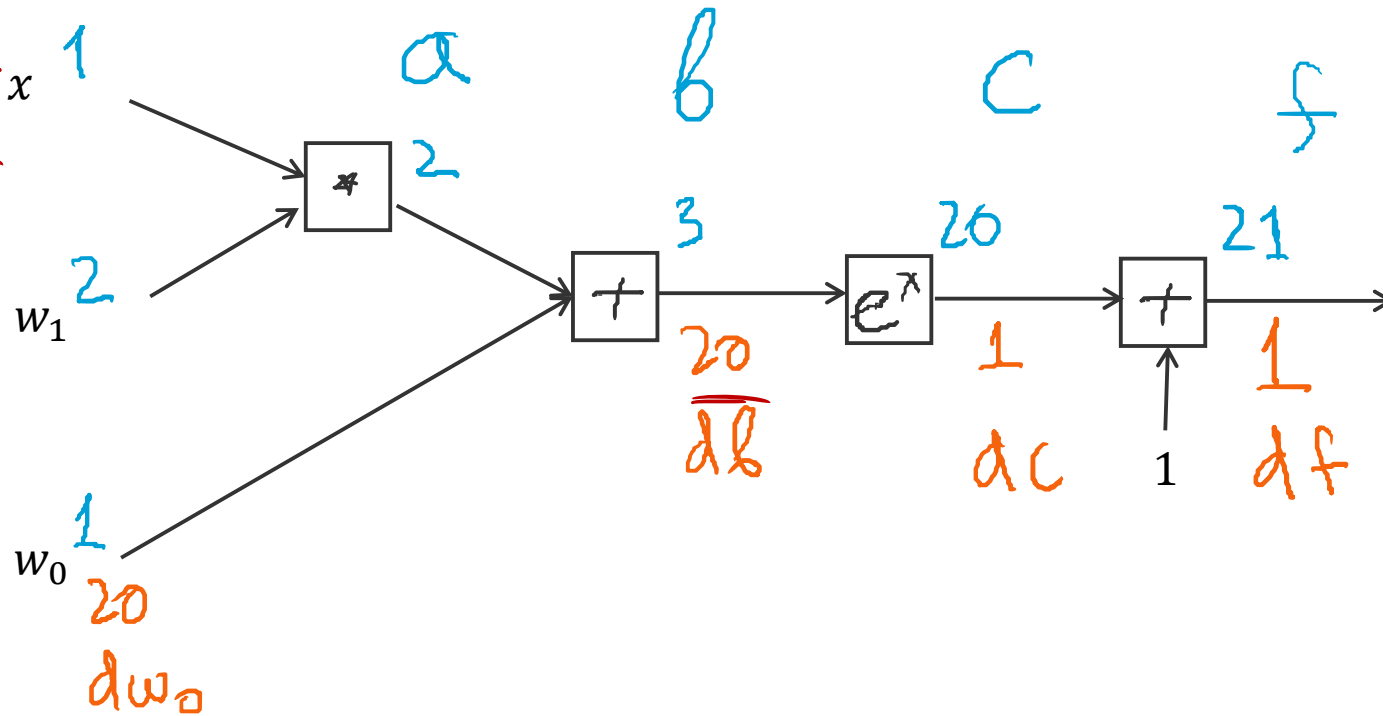
$$f(g(x)) \quad \frac{df}{dx} = \frac{df}{dg} \frac{dg}{dx}$$

$$f(x, w) = 1 + e^{w_1 x + w_0}$$

$$\frac{df}{da} = \frac{df}{db} \cdot \frac{db}{da}$$

"20"
"1"

$$b = a + w_0$$



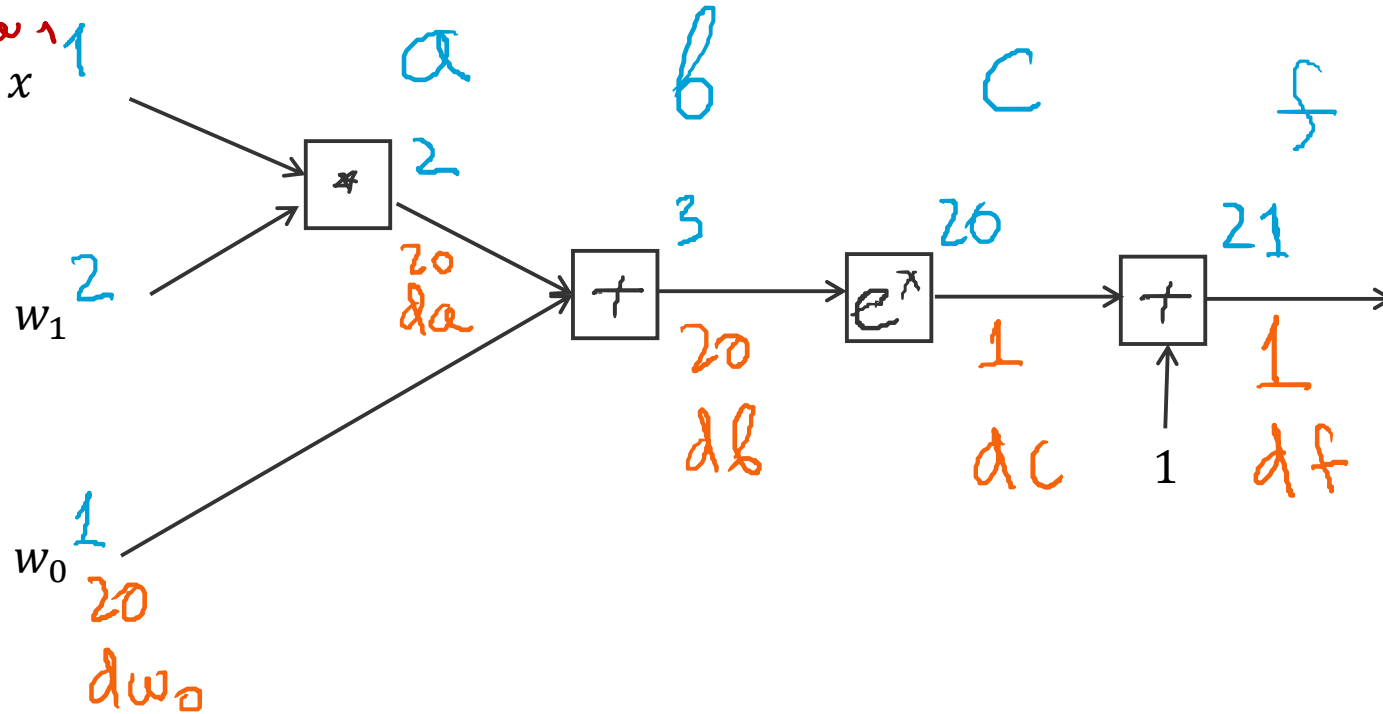
Тренируемся на кошках

$$f(g(x)) \quad \frac{df}{dx} = \frac{df}{dg} \frac{dg}{dx}$$

$$f(x, w) = 1 + e^{w_1 x + w_0}$$

$$\frac{df}{dw_1} = \frac{df}{da} \cdot \frac{da}{dw_1}$$

$a = w_1 \cdot x$



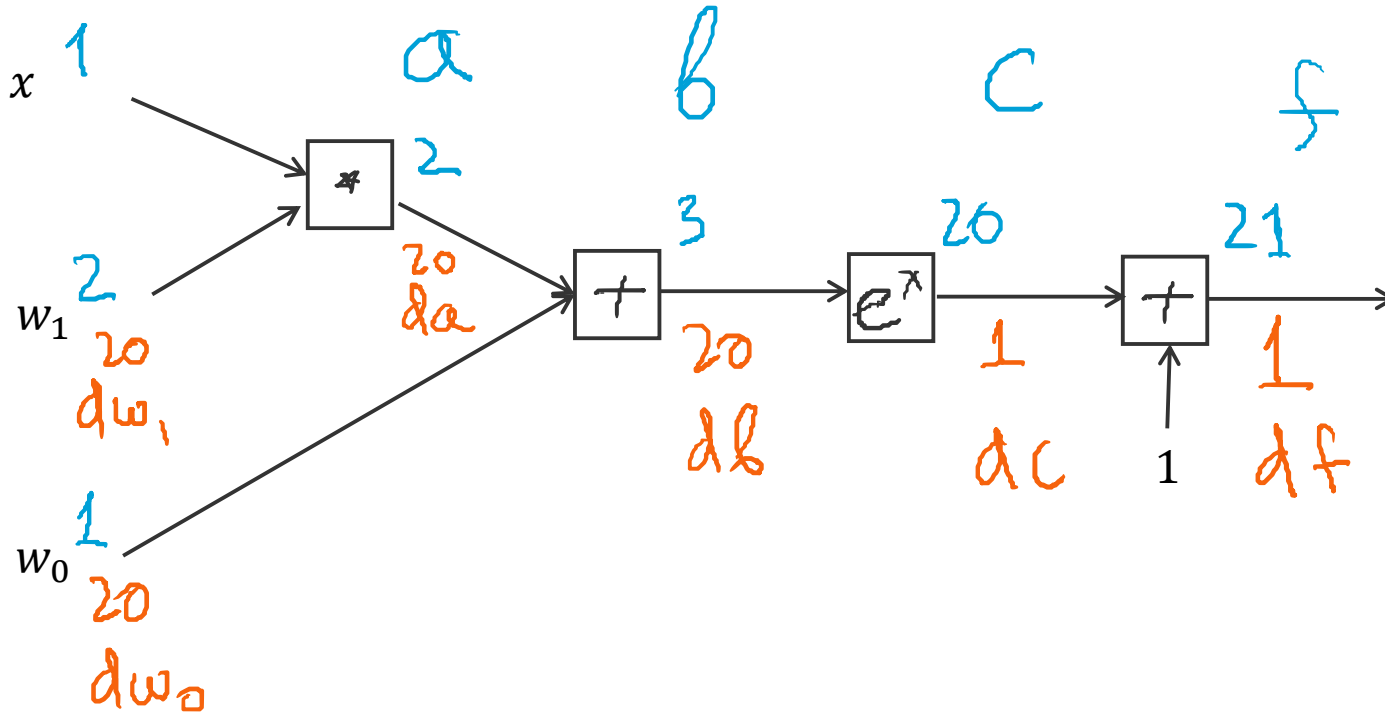
Тренируемся на кошках

$$f(g(x)) \quad \frac{df}{dx} = \frac{df}{dg} \frac{dg}{dx}$$

$$f(x, w) = 1 + e^{w_1 x + w_0}$$

$$\frac{dz}{dx} = \frac{dz}{da} \cdot \frac{da}{dx}$$

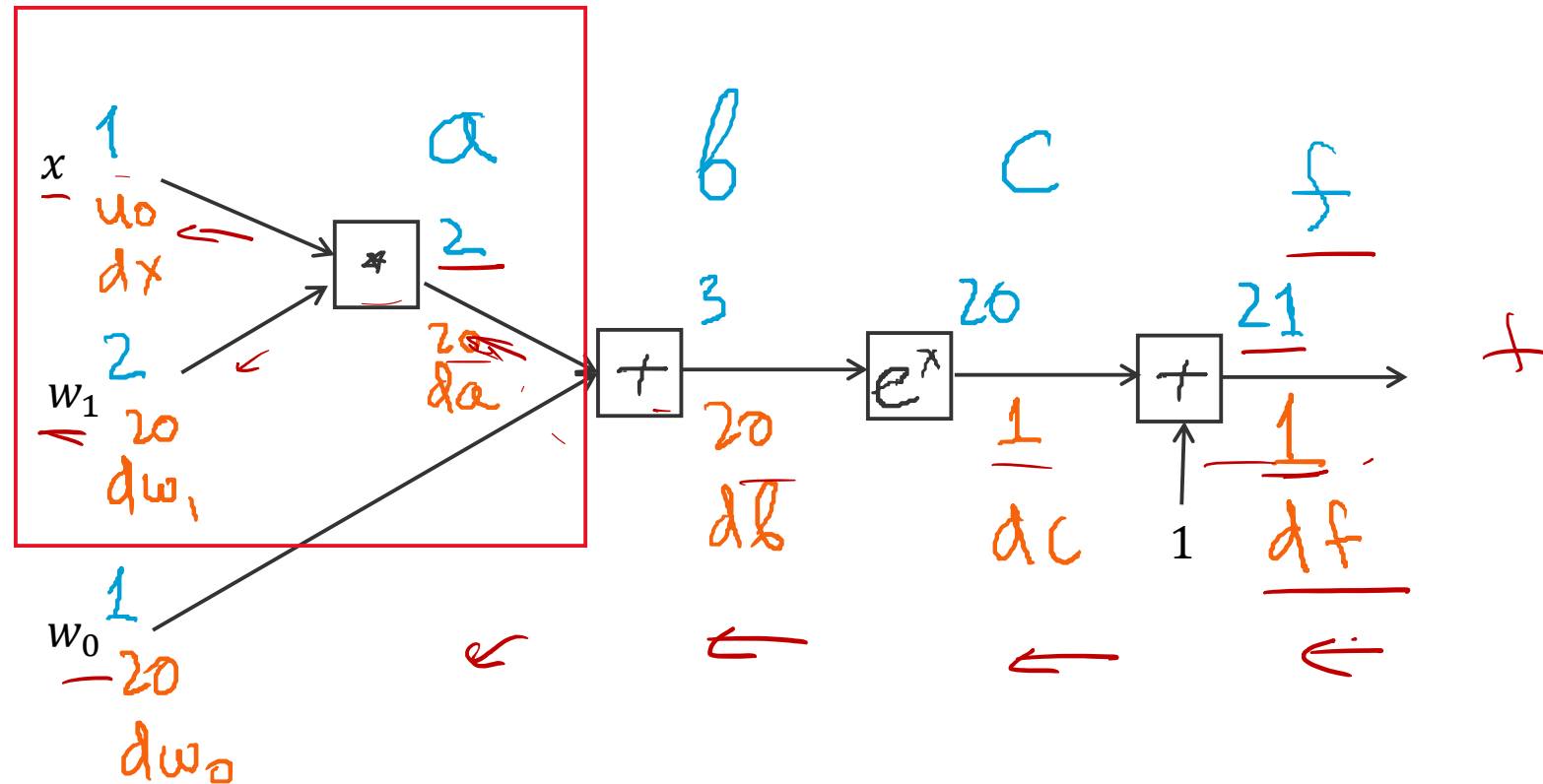
$$a = x \cdot w_1$$



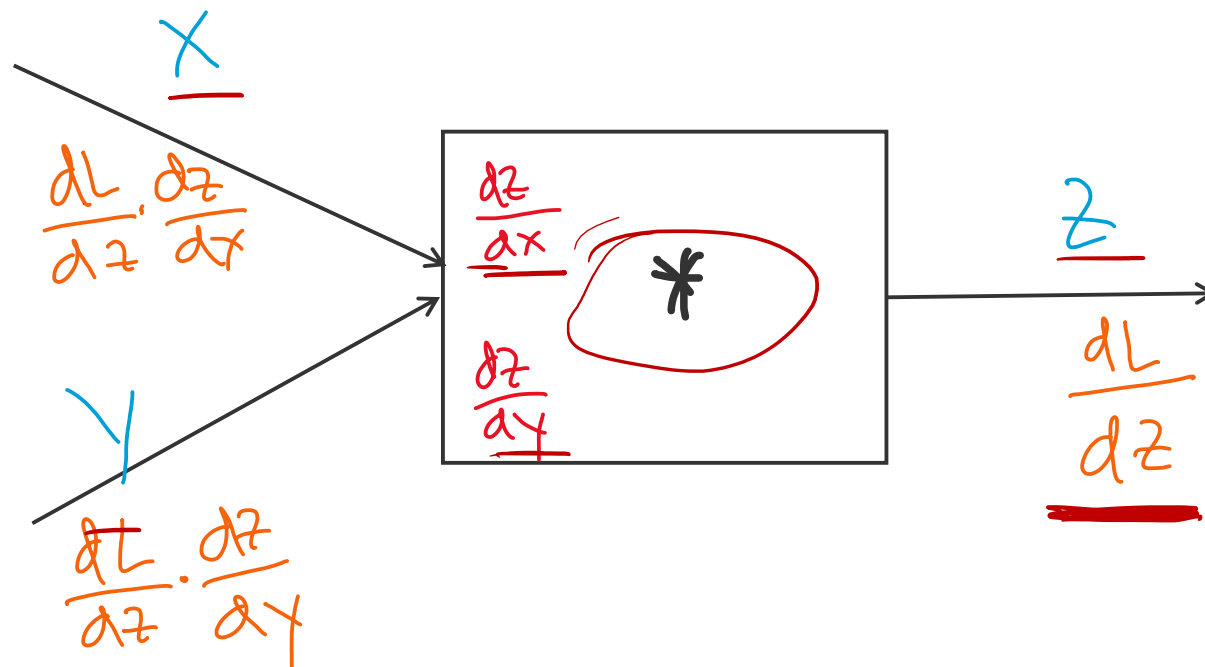
Обратное распространение ошибки Backpropagation

$$f(g(x)) \quad \frac{df}{dx} = \frac{df}{dg} \frac{dg}{dx}$$

$$f(x, w) = \underline{1 + e^{w_1 x + w_0}}$$



Общая схема вычисления градиента



$\leftarrow L$

В коде

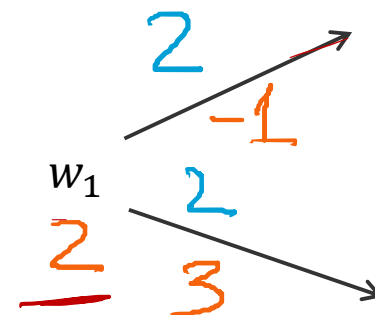
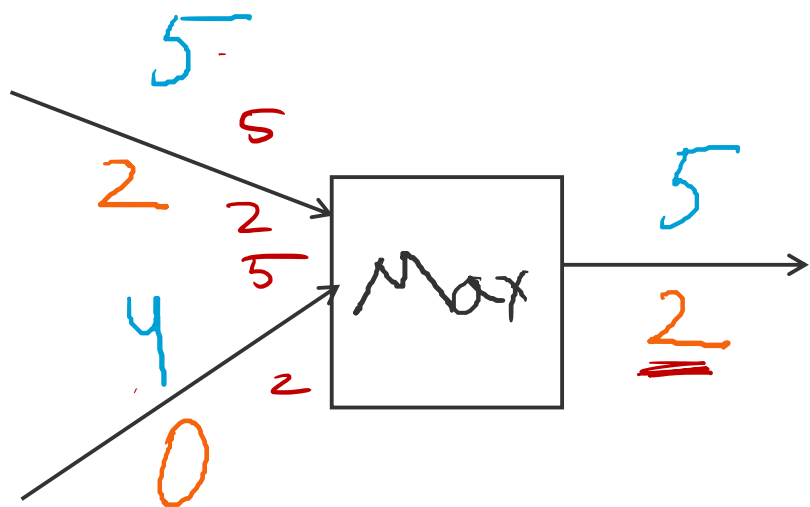
forward

```
x = 1
w1 = 2
w0 = 1
a = x*w1
b = a + w0
c = np.exp(b)
f = 1 + c
```

backward

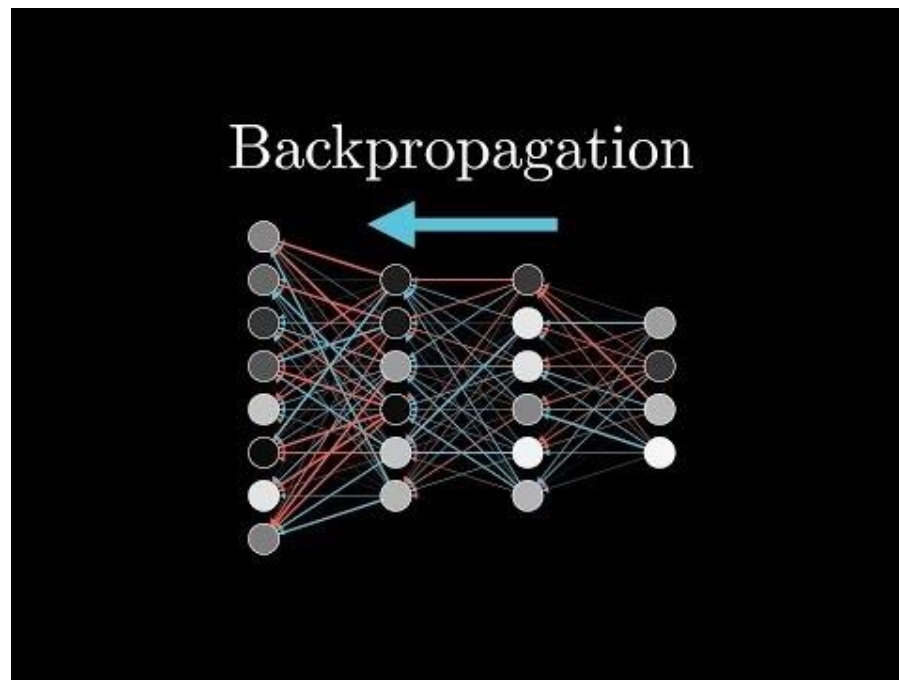
```
df = 1
dc = df
db = np.exp(b)*dc
dw0 = db
da = db
dw1 = x*da
dx = w1*da
```

Уточнения





Все еще ничего не понятно!!!!



3 blue 1 brown

<https://www.youtube.com/watch?v=llg3gGewQ5U>

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