Оптимизация в нейронных сетях.

База

Обучение NN и параллельные вычисления

$$L(W, X, y) = \frac{1}{N} \sum_{i=1}^{N} l(W, x_i, y_i) \rightarrow \min_{W \in \mathbb{R}^p}$$

ℹ️ https://developer.ibm.com/exchanges/data/all/wikitext-103/ ; https://arxiv.org/pdf/1409.0575.pdf

Обучение NN и параллельные вычисления

Функция потерь (меньше - лучше)

$$L(W, X, y) = \frac{1}{N} \sum_{i=1}^{N} l(W, x_i, y_i) \rightarrow \min_{W \in \mathbb{R}^p}$$

ℹ️ https://developer.ibm.com/exchanges/data/all/wikitext-103/ ; https://arxiv.org/pdf/1409.0575.pdf

Обучение NN и параллельные вычисления

Функция потерь (меньше - лучше)

Потери на одном объекте из обучающей выборки

$$L(W, X, y) = \frac{1}{N} \sum_{i=1}^{N} l(W, x_i, y_i) \rightarrow \min_{W \in \mathbb{R}^p}$$

ℹ️ https://developer.ibm.com/exchanges/data/all/wikitext-103/ ; https://arxiv.org/pdf/1409.0575.pdf

Обучение NN и параллельные вычисления

Функция потерь (меньше - лучше)

Потери на одном объекте из обучающей выборки

$$L(W, X, y) = \frac{1}{N} \sum_{i=1}^{N} l(W, x_i, y_i) \rightarrow \min_{W \in \mathbb{R}^p}$$

Его метка

Объект из обучающей выборки

ℹ️ https://developer.ibm.com/exchanges/data/all/wikitext-103/ ; https://arxiv.org/pdf/1409.0575.pdf

Обучение NN и параллельные вычисления

Размер обучающей выборки
ImageNet ≈ $1.4 \cdot 10^7$
WikiText ≈ 10^8

Потери на одном объекте из обучающей выборки

Функция потерь (меньше - лучше)

$$L(W, X, y) = \frac{1}{N} \sum_{i=1}^{N} l(W, x_i, y_i) \to \min_{W \in \mathbb{R}^p}$$

Его метка

Объект из обучающей выборки

ℹ️ https://developer.ibm.com/exchanges/data/all/wikitext-103/ ; https://arxiv.org/pdf/1409.0575.pdf

Обучение NN и параллельные вычисления

Размер обучающей выборки
ImageNet ≈ $1.4 \cdot 10^7$
WikiText ≈ 10^8

Потери на одном объекте из обучающей выборки

Функция потерь (меньше - лучше)

$$L(W, X, y) = \frac{1}{N} \sum_{i=1}^{N} l(W, x_i, y_i) \to \min_{W \in \mathbb{R}^p}$$

Его метка

Веса модели, которые нужно подобрать
НО КАК?

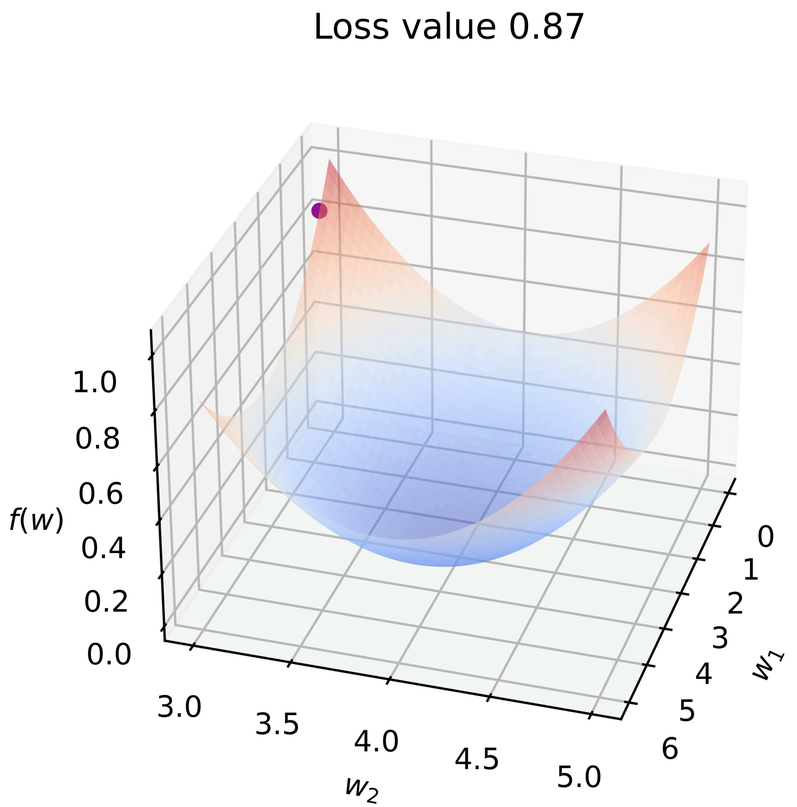
Объект из обучающей выборки

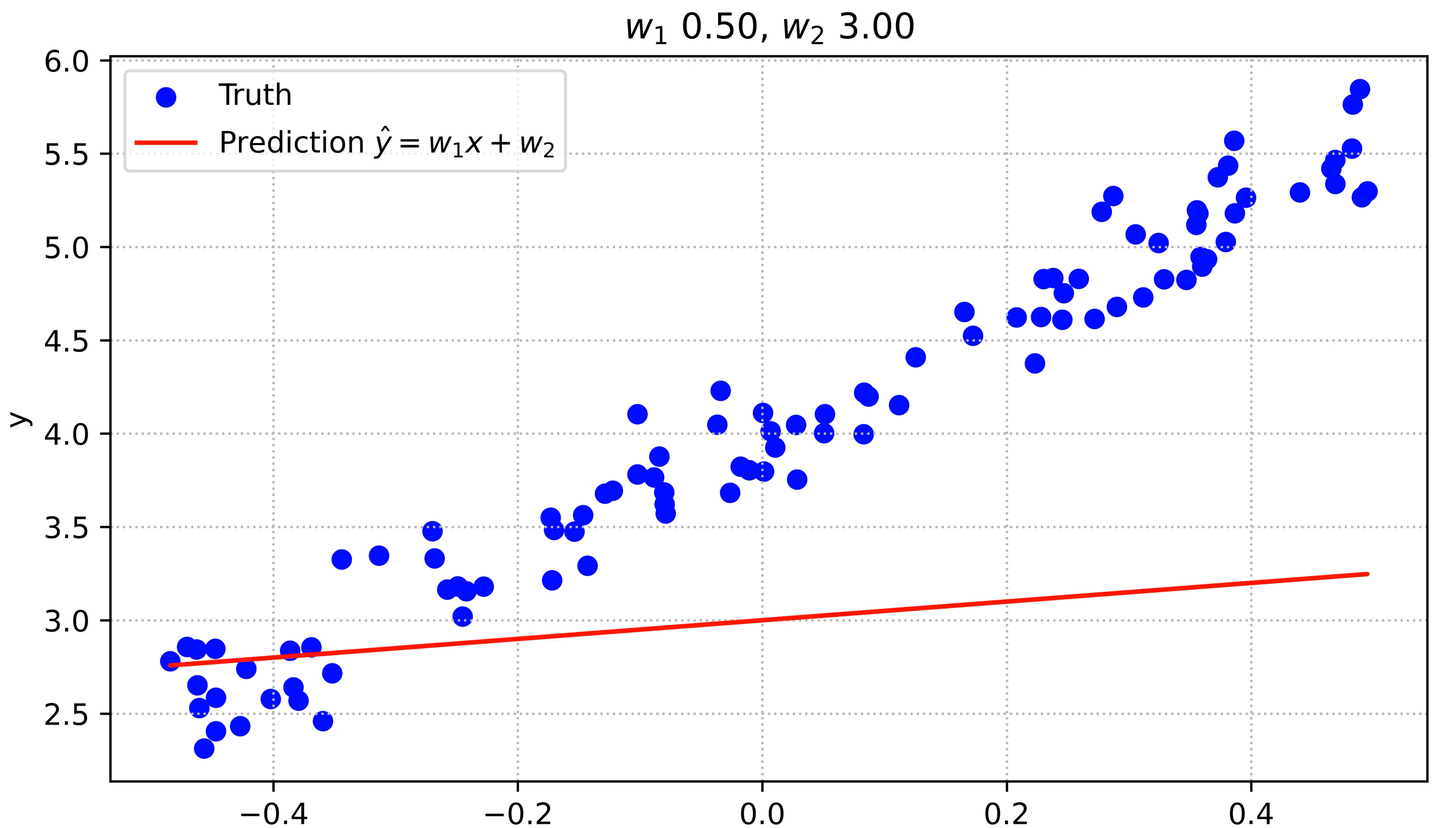
ℹ️ https://developer.ibm.com/exchanges/data/all/wikitext-103/ ; https://arxiv.org/pdf/1409.0575.pdf

Обучение NN и параллельные вычисления

Метод градиентного спуска (GD)

$$W_{k+1} = W_k - \alpha \nabla_W L(W_k)$$





https://colab.research.google.com/github/MerkulovDaniil/optim/blob/master/assets/Notebooks/GD_2d_visualization.ipynb

Обучение NN и параллельные вычисления

Метод градиентного спуска (GD)

$$W_{k+1} = W_k - \alpha \nabla_W L(W_k)$$

$$L(W, X, y) = \frac{1}{N} \sum_{i=1}^{N} l(W, x_i, y_i) \rightarrow \min_{W \in \mathbb{R}^p}$$

ℹ️ https://colab.research.google.com/github/MerkulovDaniil/optim/blob/master/assets/Notebooks/GD_2d_visualization.ipynb

Обучение NN и параллельные вычисления

Метод градиентного спуска (GD)

$$W_{k+1} = W_k - \alpha \nabla_W L(W_k)$$

$$L(W, X, y) = \frac{1}{N} \sum_{i=1}^{N} l(W, x_i, y_i) \rightarrow \min_{W \in \mathbb{R}^p}$$

$$\nabla_W L(W_k) = \frac{1}{N} \sum_{i=1}^{N} \nabla_W l(W_k, x_i, y_i)$$

ℹ️ https://colab.research.google.com/github/MerkulovDaniil/optim/blob/master/assets/Notebooks/GD_2d_visualization.ipynb

Обучение NN и параллельные вычисления

Метод градиентного спуска (GD)

$$W_{k+1} = W_k - \alpha \nabla_W L(W_k)$$

$$L(W, X, y) = \frac{1}{N} \sum_{i=1}^{N} l(W, x_i, y_i) \rightarrow \min_{W \in \mathbb{R}^p}$$

$$\nabla_W L(W_k) = \frac{1}{N} \sum_{i=1}^{N} \nabla_W l(W_k, x_i, y_i)$$

Тяжело считать при большом N 🤔

ℹ️ https://colab.research.google.com/github/MerkulovDaniil/optim/blob/master/assets/Notebooks/GD_2d_visualization.ipynb

Обучение NN и параллельные вычисления

Метод стохастического градиентного спуска (SGD)

$$W_{k+1} = W_k - \alpha g_k$$

$$\nabla_W L(W_k) = \frac{1}{N} \sum_{i=1}^{N} \nabla_W l(W_k, x_i, y_i)$$

ℹ️ https://colab.research.google.com/github/MerkulovDaniil/optim/blob/master/assets/Notebooks/GD_2d_visualization.ipynb

Обучение NN и параллельные вычисления

Метод стохастического градиентного спуска (SGD)

$$W_{k+1} = W_k - \alpha g_k$$

$$\nabla_W L(W_k) = \frac{1}{N} \sum_{i=1}^{N} \nabla_W l(W_k, x_i, y_i)$$

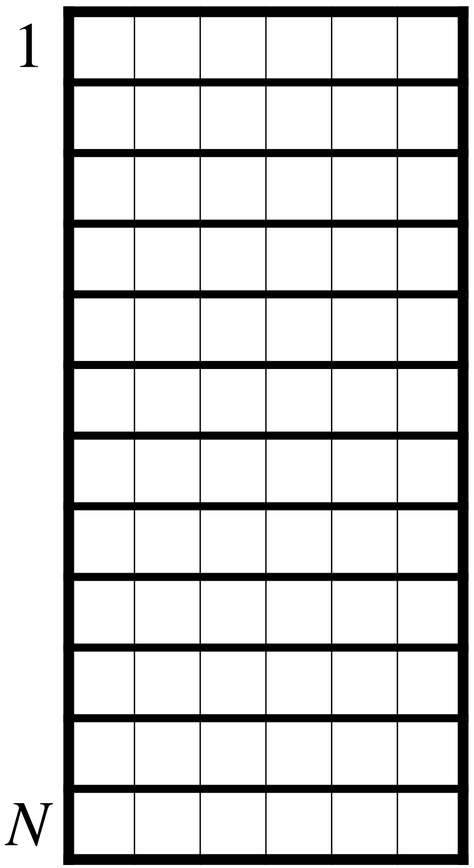
$$g_k = \frac{1}{b} \sum_{i=1}^{b} \nabla_W l(W_k, x_{j_i}, y_{j_i})$$

$$b \ll N$$

ℹ️ https://colab.research.google.com/github/MerkulovDaniil/optim/blob/master/assets/Notebooks/GD_2d_visualization.ipynb

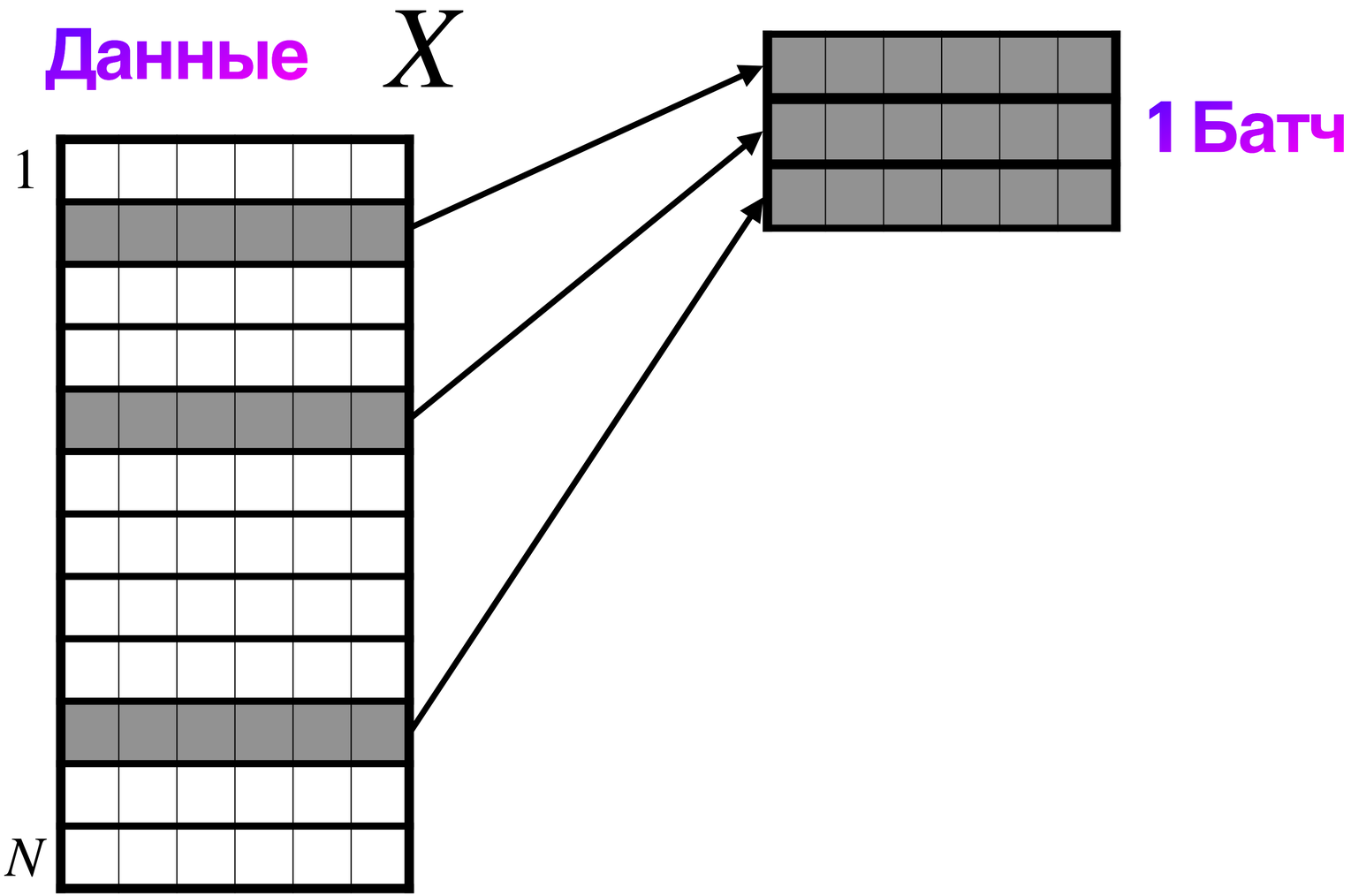
Обучение NN и параллельные вычисления

Данные X



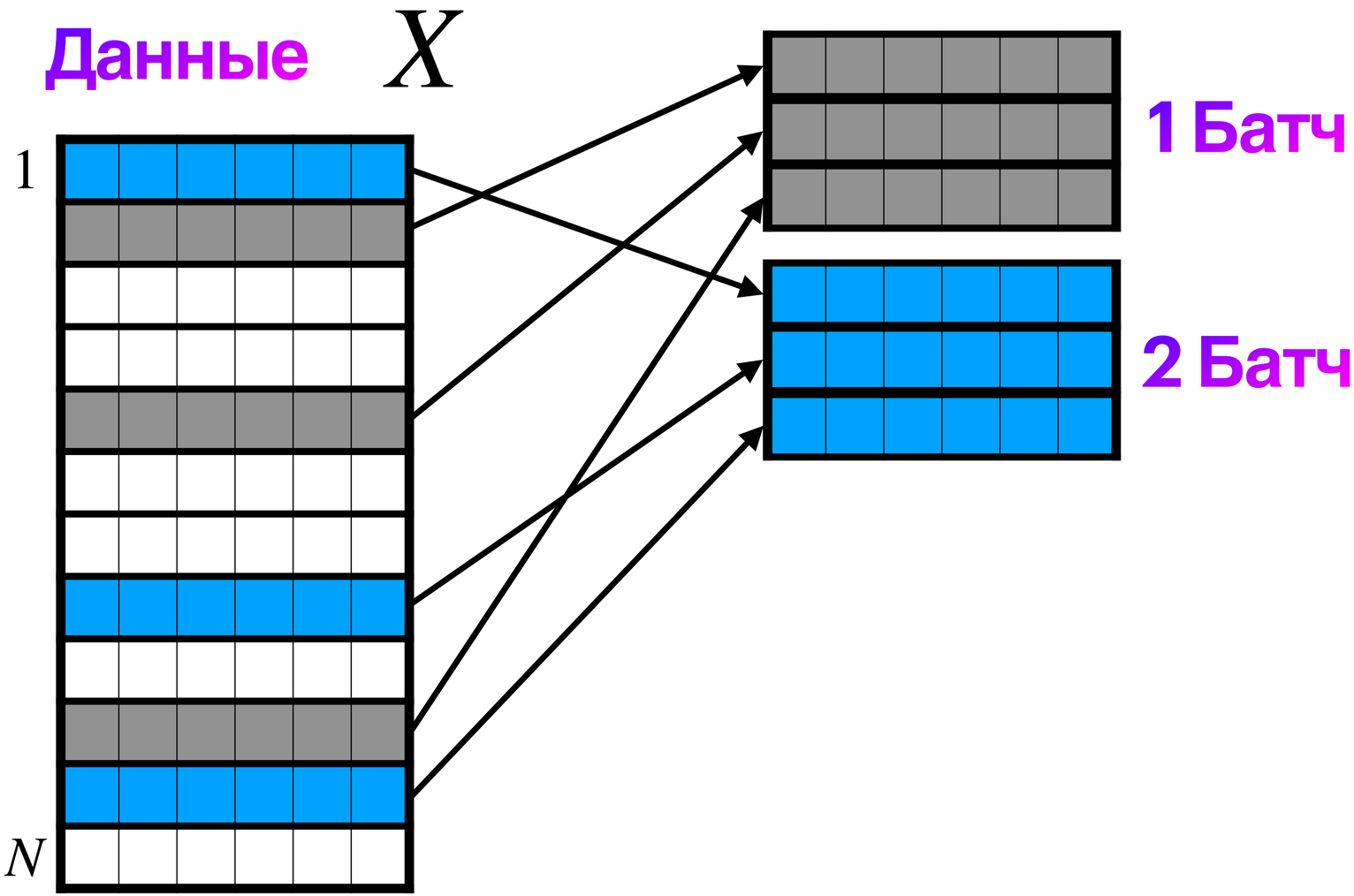
ℹ️ https://colab.research.google.com/github/MerkulovDaniil/optim/blob/master/assets/Notebooks/GD_2d_visualization.ipynb

Обучение NN и параллельные вычисления



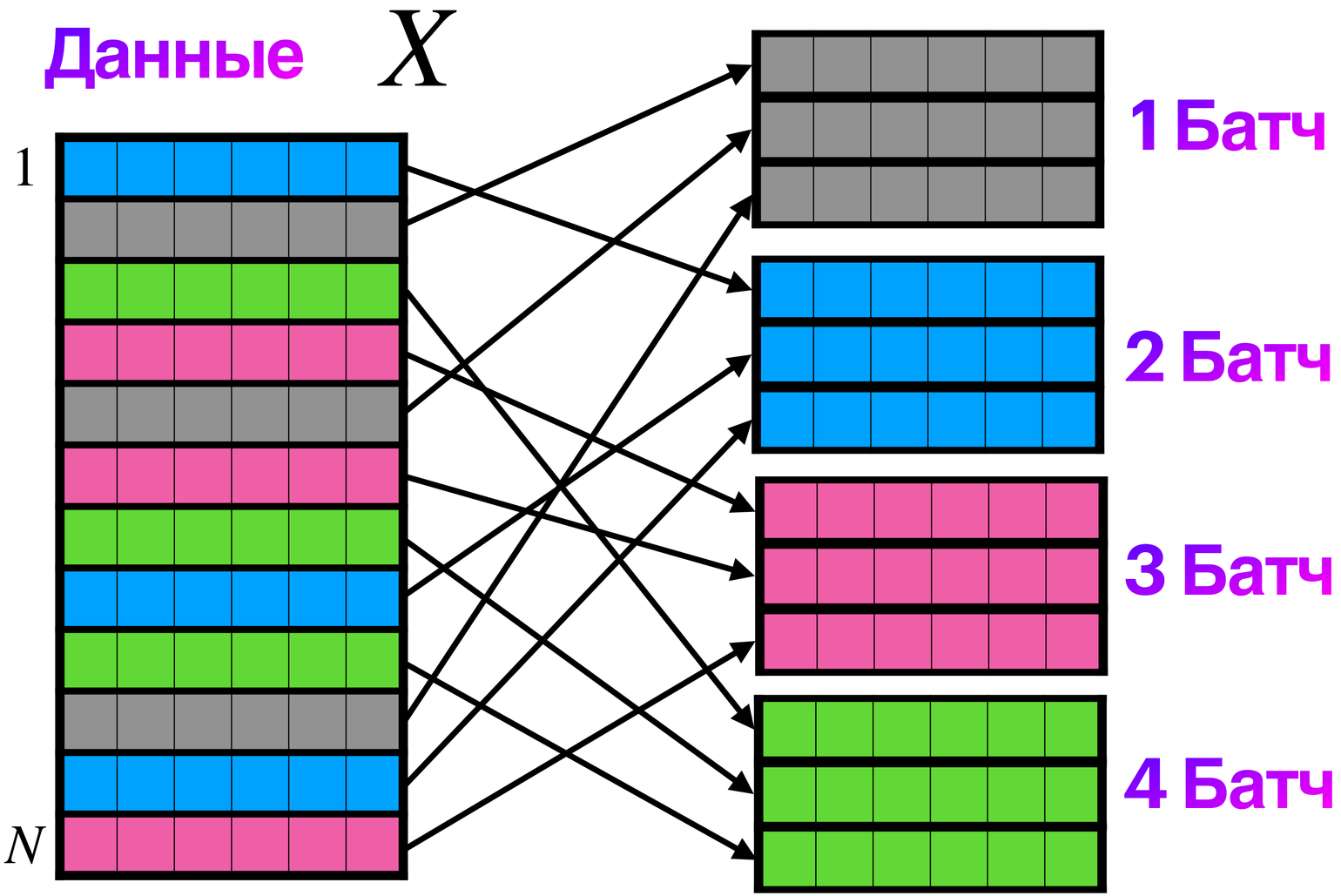
https://colab.research.google.com/github/MerkulovDaniil/optim/blob/master/assets/Notebooks/GD_2d_visualization.ipynb

Обучение NN и параллельные вычисления



https://colab.research.google.com/github/MerkulovDaniil/optim/blob/master/assets/Notebooks/GD_2d_visualization.ipynb

Обучение NN и параллельные вычисления



https://colab.research.google.com/github/MerkulovDaniil/optim/blob/master/assets/Notebooks/GD_2d_visualization.ipynb

Свое железо для обучения - дорого

	ARCH	VRAM	FP32 TFLOPS	ENERGY	ENPRICE	PPRICE
RTX 2080 Ti	Turing	11	13.5	250	500 €	450 €
RTX 2080 Super	Turing	8	11.2	215	430 €	350 €
RTX 2070 Super	Turing	8	9.1	175	350 €	300 €
RTX 4090	Ada Lovelace	24	82.6	450	900 €	1,900 €
RTX 4070 Ti	Ada Lovelace	12	40.1	285	570 €	1,000 €
RTX 3070 Ti	Ampere	8	21.75	290	580 €	600 €
RTX 4080	Ada Lovelace	16	48.7	320	640 €	1,300 €
RTX 3080 Ti	Ampere	12	34.1	350	700 €	1,100 €
RTX 3090 Ti	Ampere	24	40	450	900 €	1,500 €
RTX A4000	Ampere	16	19.1	140	280 €	1,000 €
RTX 4000 Ada	Ada Lovelace	20	19.2	70	140 €	1,250 €
RTX A5000	Ampere	24	27.8	230	460 €	2,500 €
RTX 6000 Ada	Ada Lovelace	48	91.1	300	600 €	7,000 €
L40	Ada Lovelace	48	90.5	300	600 €	9,000 €
RTX A6000	Ampere	48	38.7	300	600 €	6,000 €
A100	Ampere	40	19.5	250	500 €	11,000 €
H100	Hopper	80	24.1	350	700 €	35,000 €

ENPRICE - оценка стоимости работы GPU на 100% в течение года

Сайт
Таблица ценами на GPU и сравнением

Теория

Стох. градиент - несмещенная оценка настоящего градиента

$$\mathbb{E} g_k = \nabla_W L(W_k)$$

ℹ️ https://colab.research.google.com/github/MerkulovDaniil/optim/blob/master/assets/Notebooks/SGD_and_variations_ISP.ipynb

SGD с постоянным шагом не сходится для выпуклой задачи

“Сходимость” для сильно выпуклой задачи:

$$\mathbb{E}[\|x_k - x^*\|^2] \leq (1 - 2\alpha_k\mu)^k R^2 + \frac{\alpha B^2}{2\mu},$$

ℹ️ https://colab.research.google.com/github/MerkulovDaniil/optim/blob/master/assets/Notebooks/SGD_and_variations_ISP.ipynb

SGD в отличие от GD может “выпрыгивать” из локальных минимумов

ℹ️ https://colab.research.google.com/github/MerkulovDaniil/optim/blob/master/assets/Notebooks/SGD_and_variations_ISP.ipynb

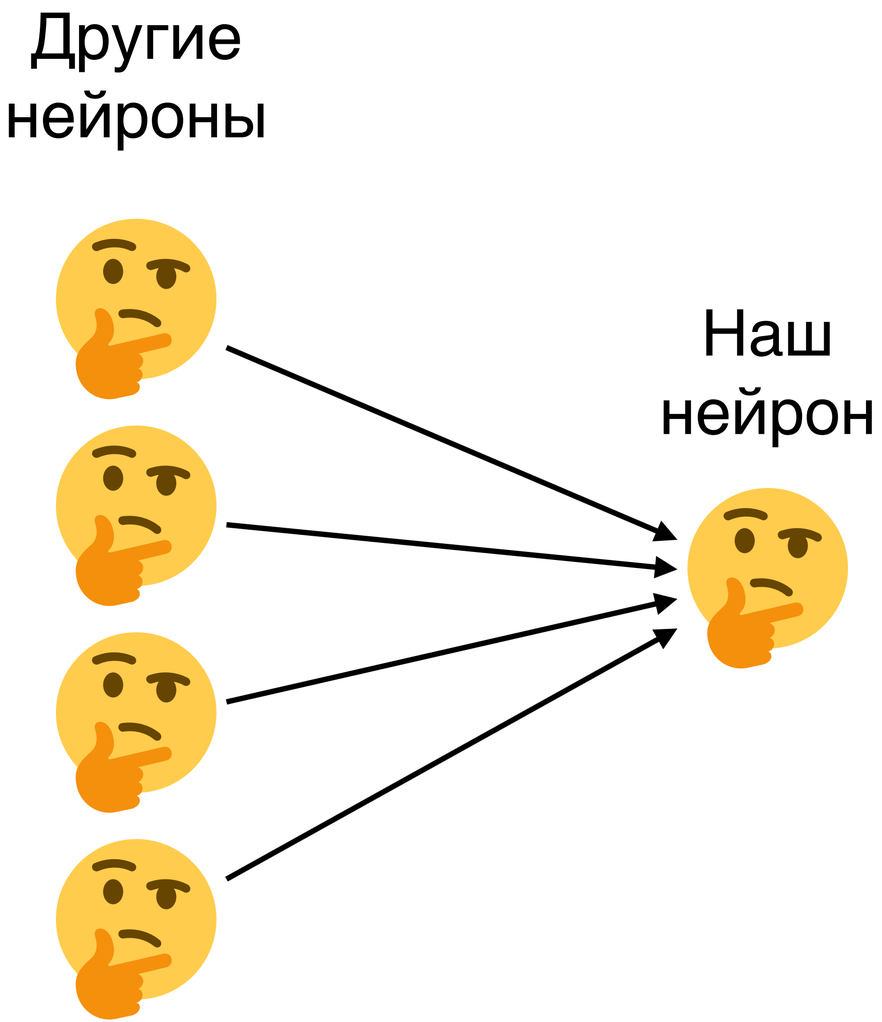
Как работает 🧠 ?

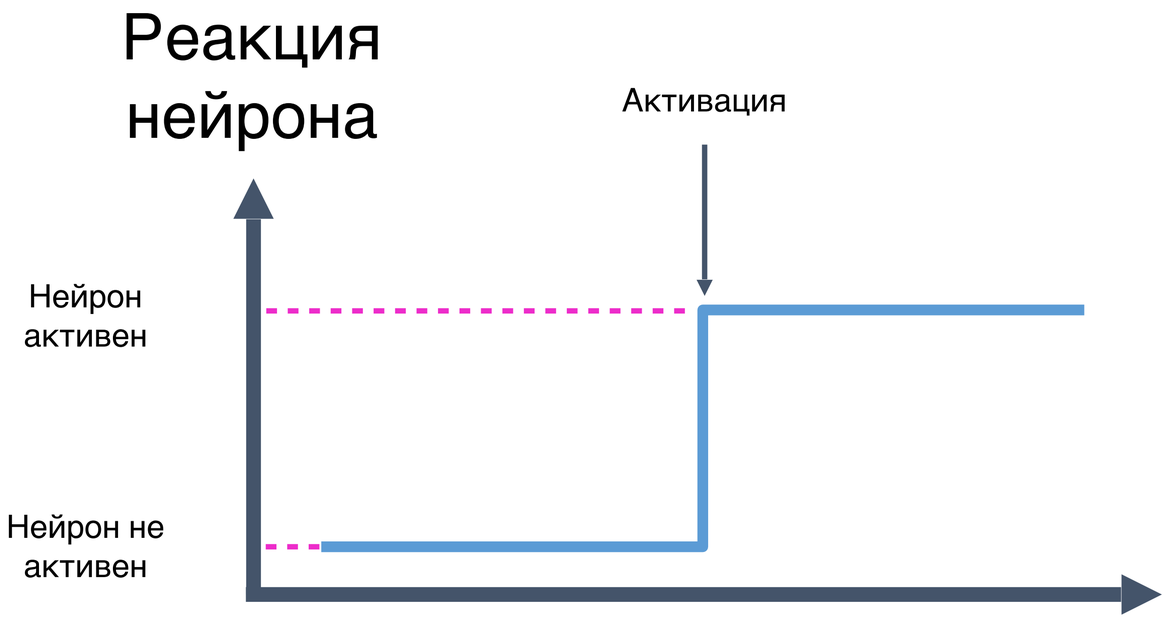
8.6×10^{10} нейронов в мозге человека

$\sim 1.5 \times 10^{14}$ связей между ними

ℹ️ https://pubmed.ncbi.nlm.nih.gov/19226510/; https://pubmed.ncbi.nlm.nih.gov/3284447/; https://doi.org/10.1016/S0531-5565(02)00151-1

Как работает 🧠 ?





Сумма входных сигналов
от других нейронов

Как работает 🤖🧠 ?

Другие нейроны

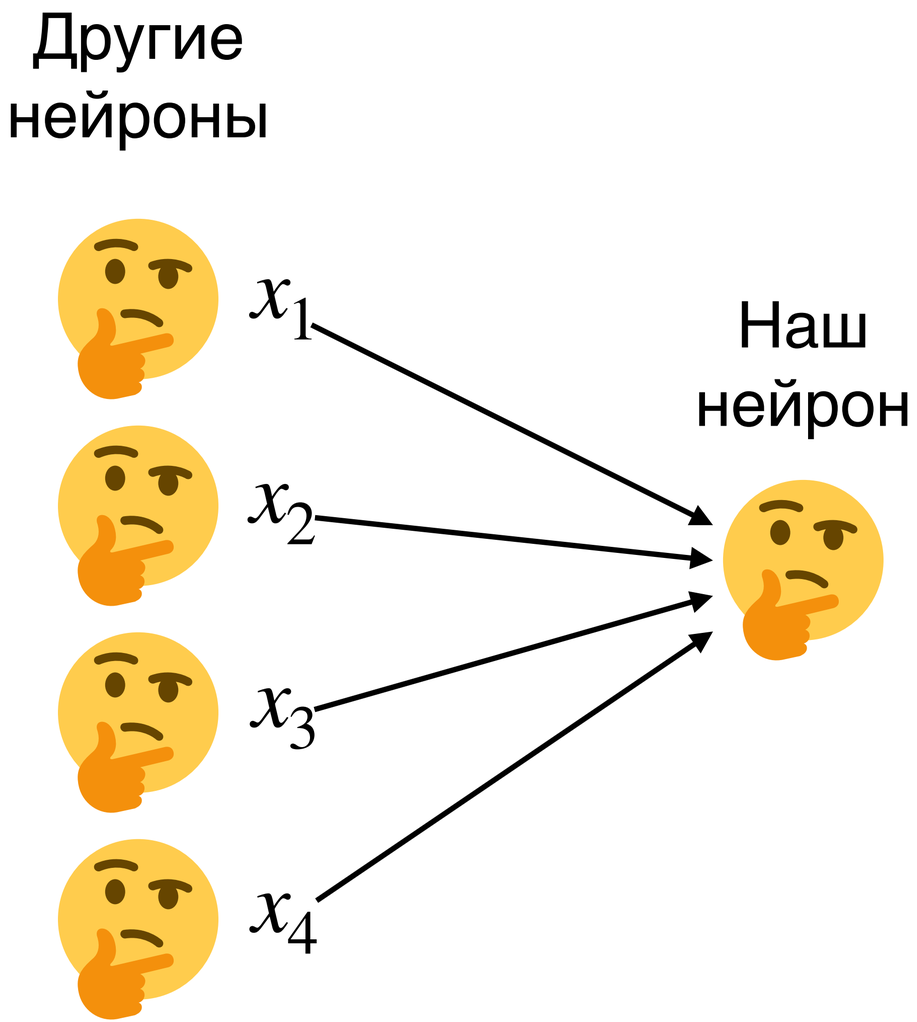
x_1

x_2

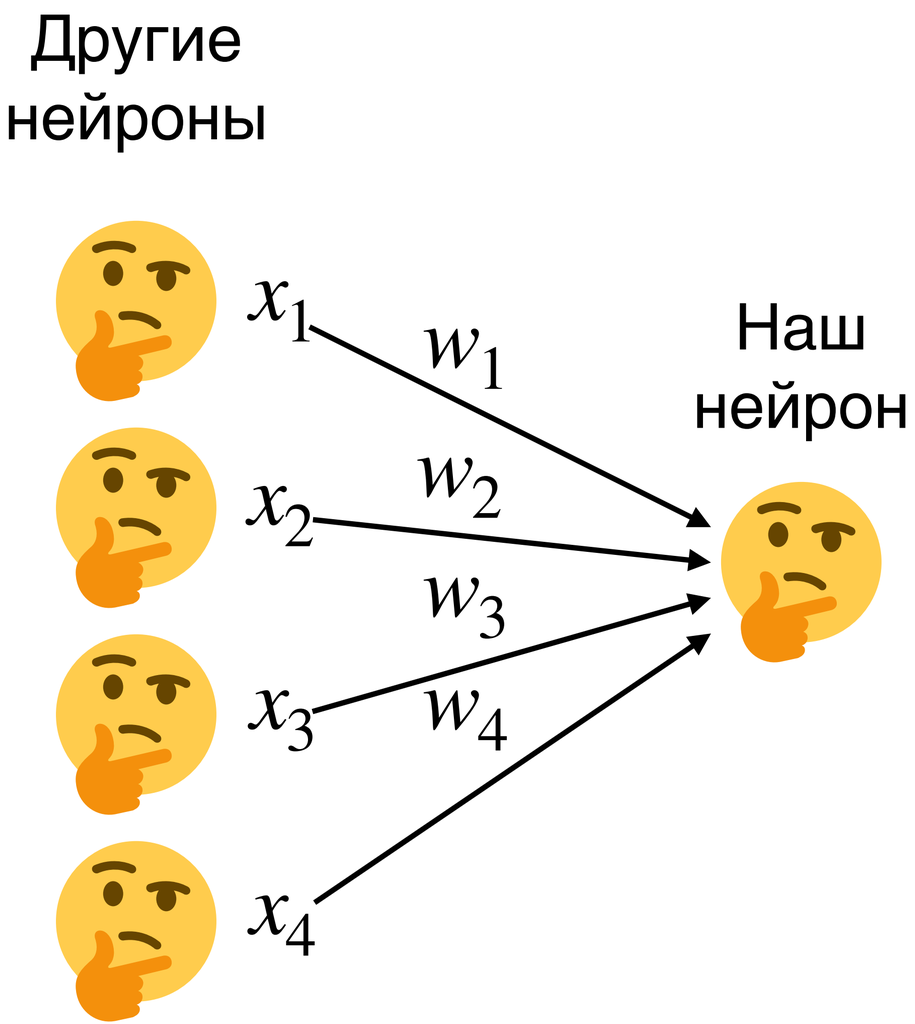
x_3

x_4

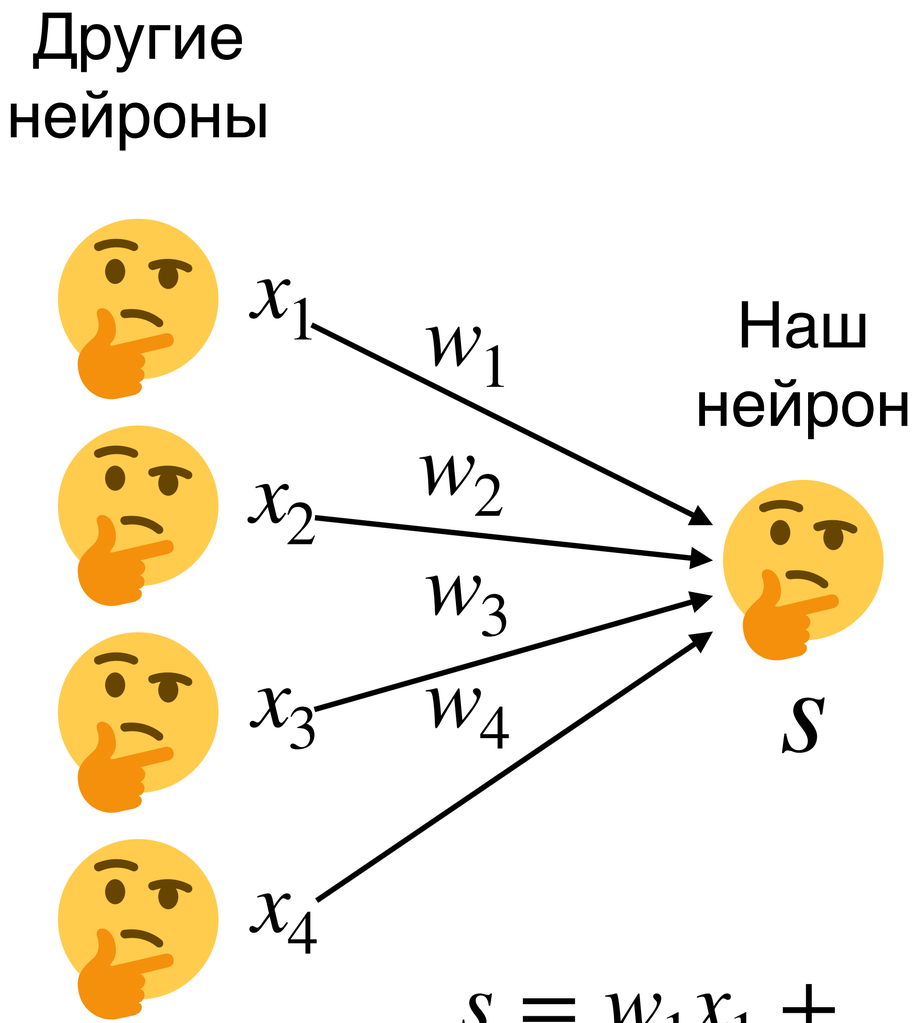
Как работает 🤖🧠 ?



Как работает 🤖🧠 ?

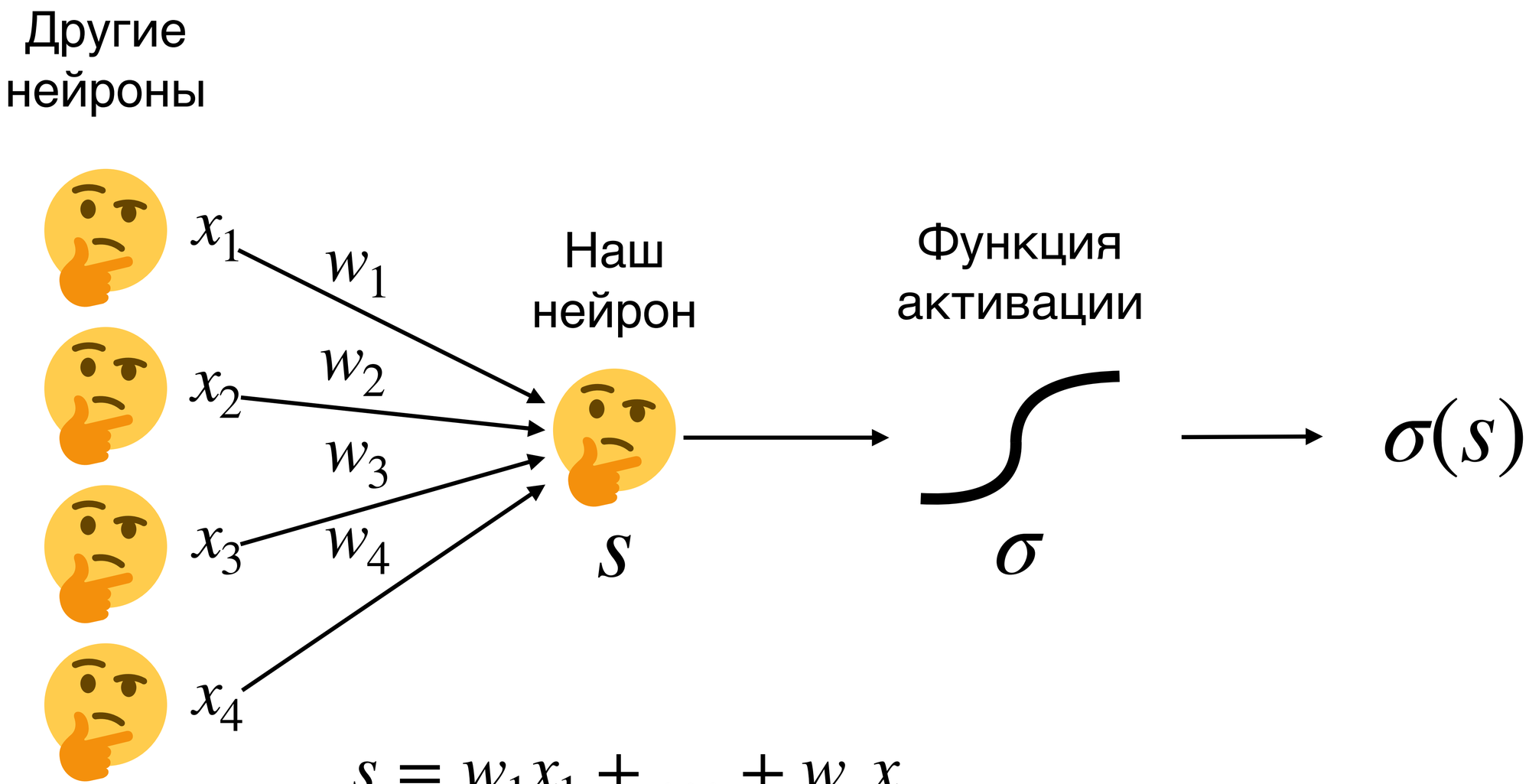


Как работает 🤖🧠 ?



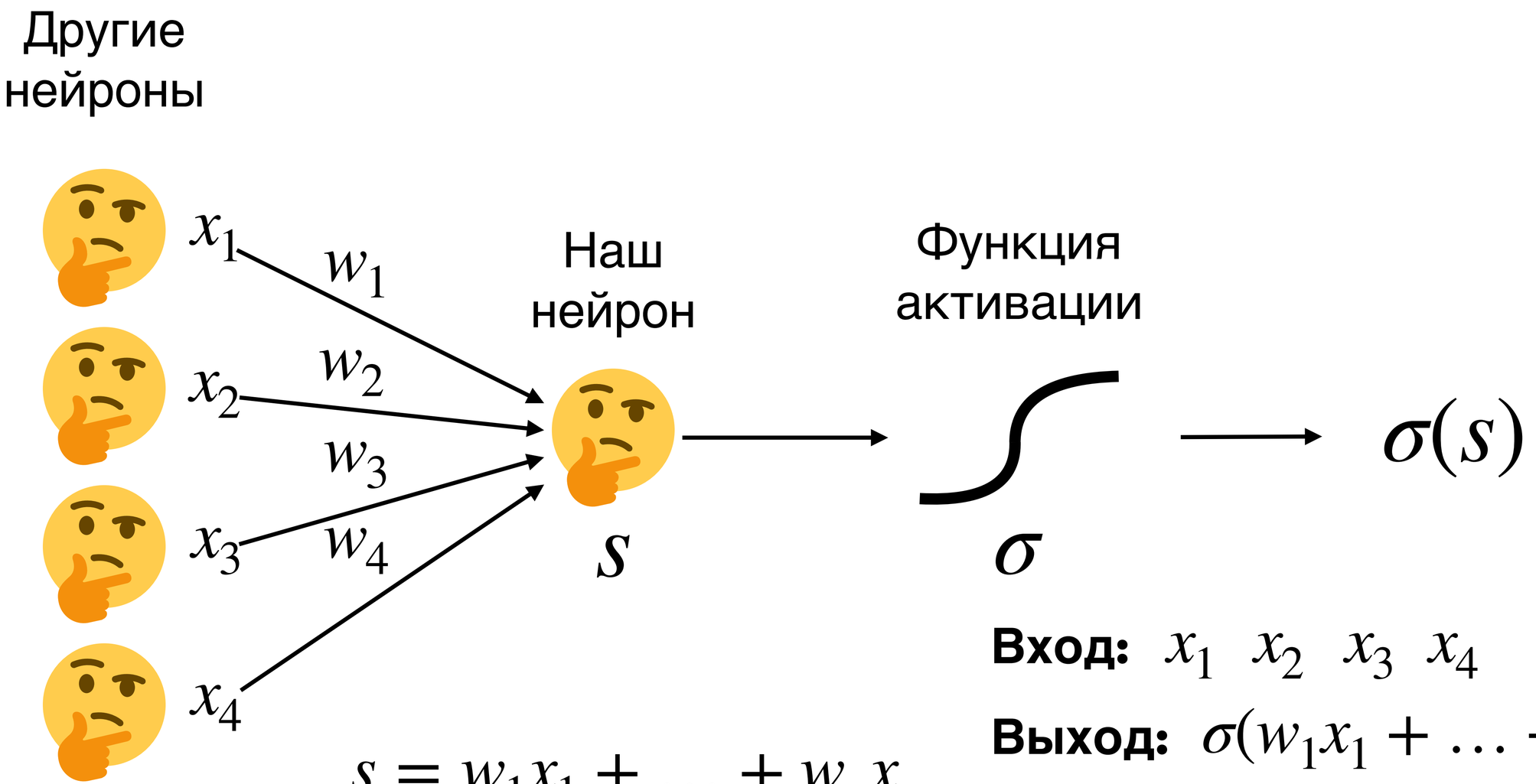
$$s = w_1 x_1 + \ldots + w_n x_n$$

Как работает 🤖🧠 ?



$$s = w_1 x_1 + \ldots + w_n x_n$$

Как работает 🤖🧠 ?



$$s = w_1 x_1 + \ldots + w_n x_n$$

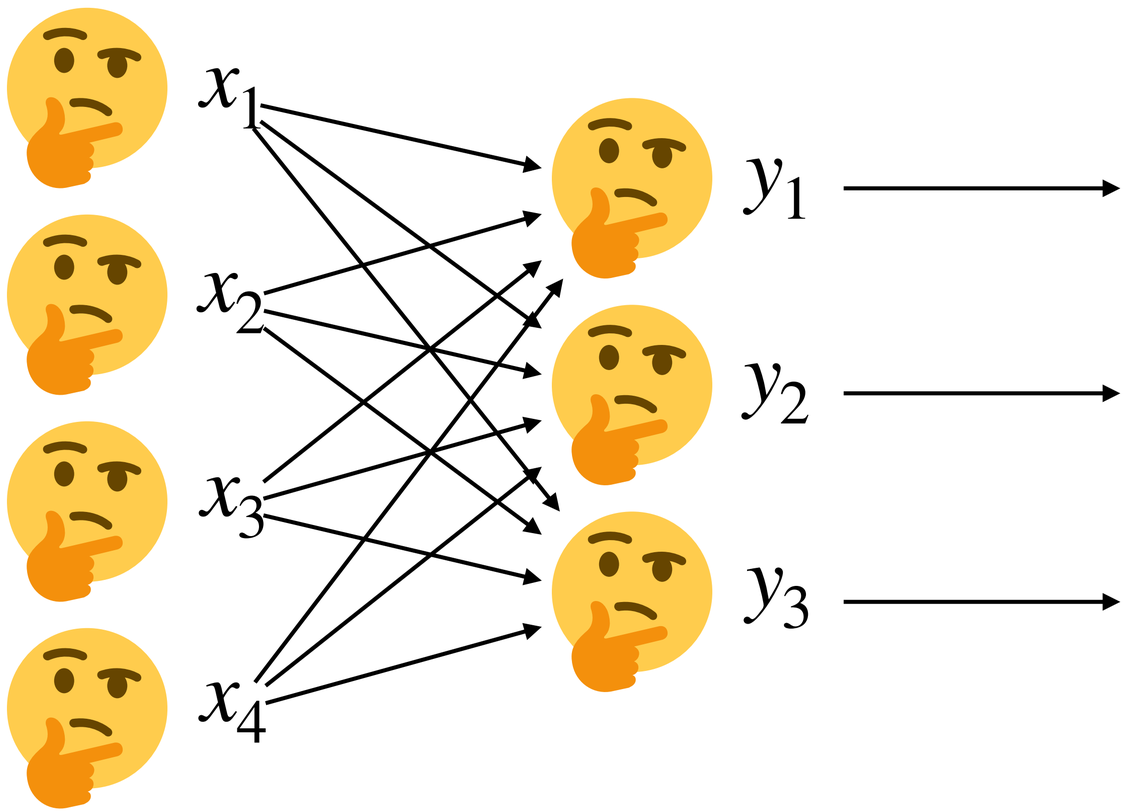
Вход: $x_1 \;\; x_2 \;\; x_3 \;\; x_4$

Выход: $\sigma(w_1 x_1 + \ldots + w_n x_n)$

Параметры: $w_1 \;\; w_2 \;\; w_3 \;\; w_4$

Как работает 🤖🧠 ?

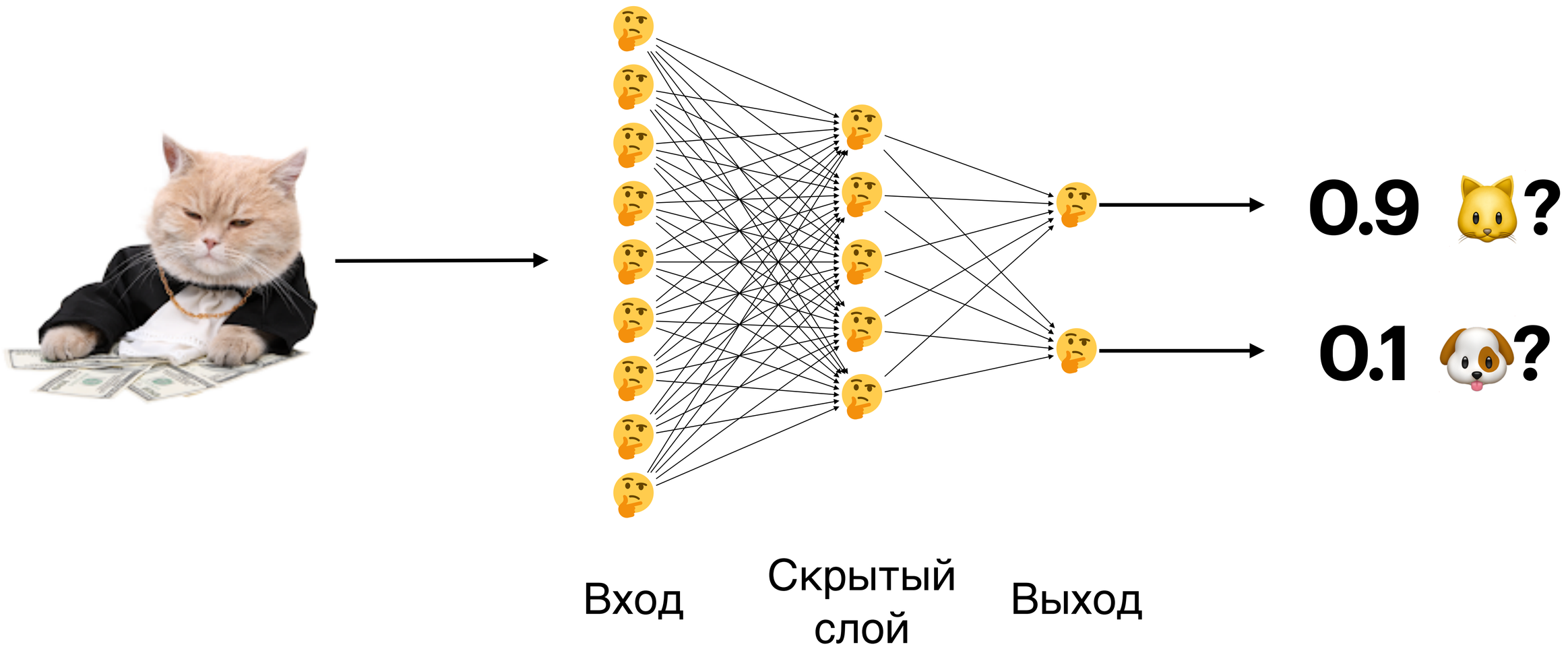
Возьмем несколько нейронов и получим полносвязный слой



$$y = \sigma(Wx)$$

3x1 3x4 4x1

Как использовать 🤖🧠 ?



Обучение нейросети

Объект
$x \in \mathbb{R}^d; y \in \mathbb{R}^c$

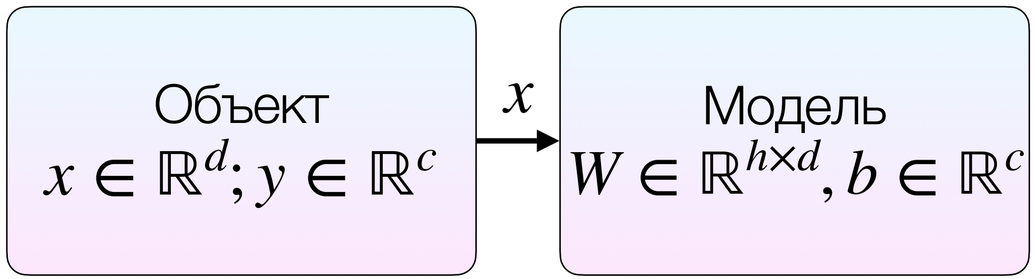
котик

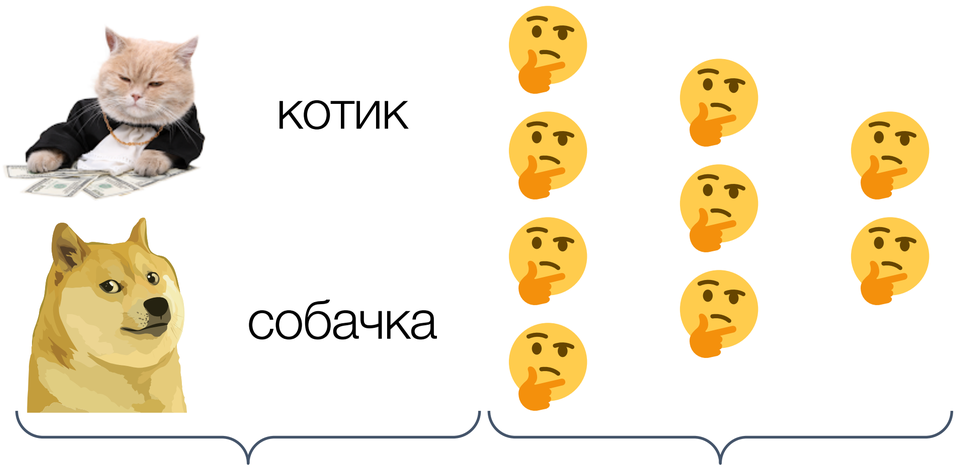
собачка

Обучающая выборка

https://colab.research.google.com/github/MerkulovDaniil/optim/blob/master/assets/Notebooks/GD_2d_visualization.ipynb

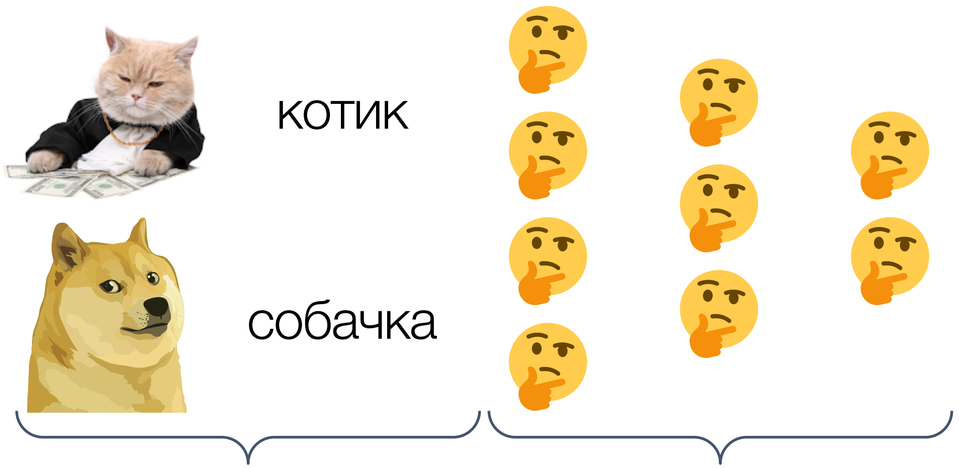
Обучение нейросети





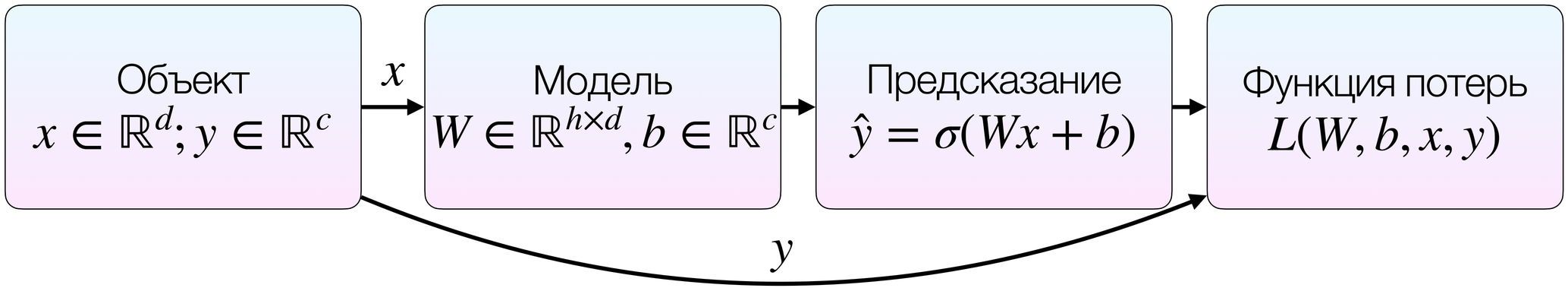
https://colab.research.google.com/github/MerkulovDaniil/optim/blob/master/assets/Notebooks/GD_2d_visualization.ipynb

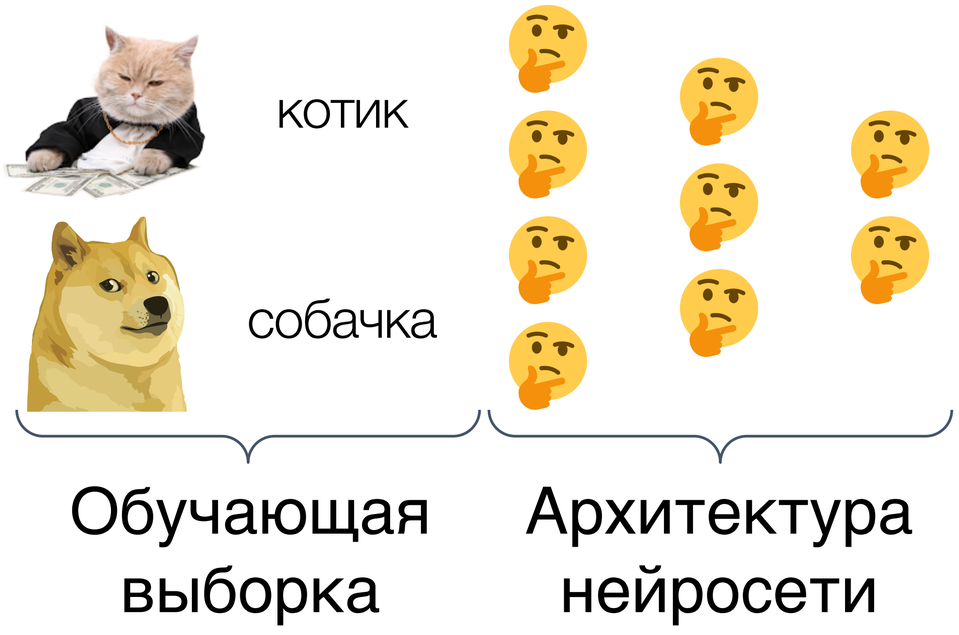
Обучение нейросети



https://colab.research.google.com/github/MerkulovDaniil/optim/blob/master/assets/Notebooks/GD_2d_visualization.ipynb

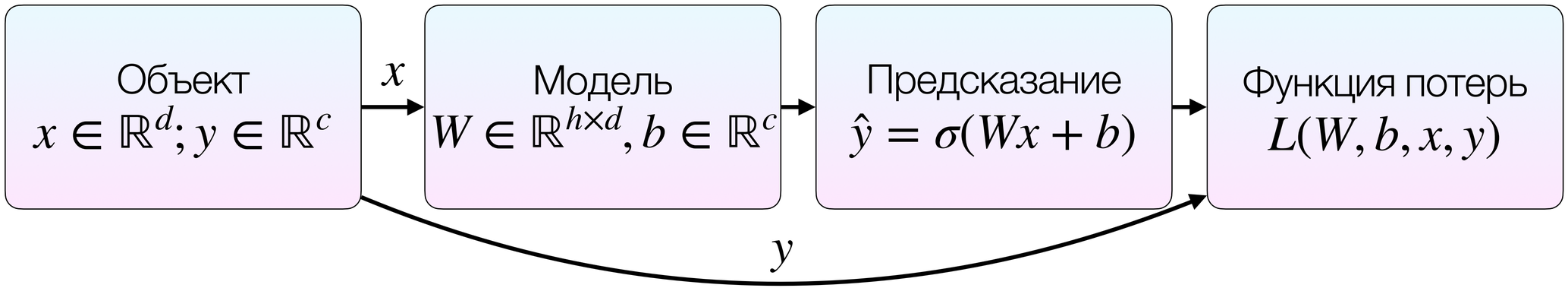
Обучение нейросети





https://colab.research.google.com/github/MerkulovDaniil/optim/blob/master/assets/Notebooks/GD_2d_visualization.ipynb

Обучение нейросети

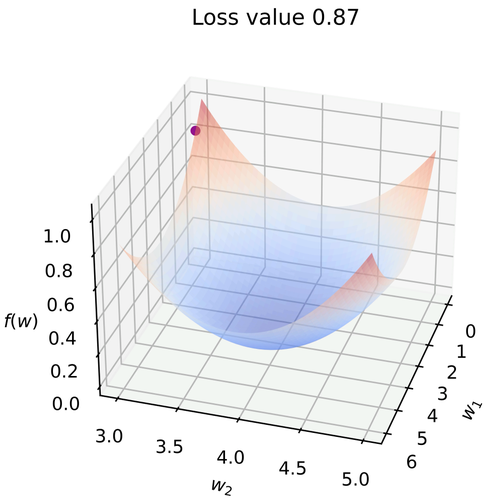


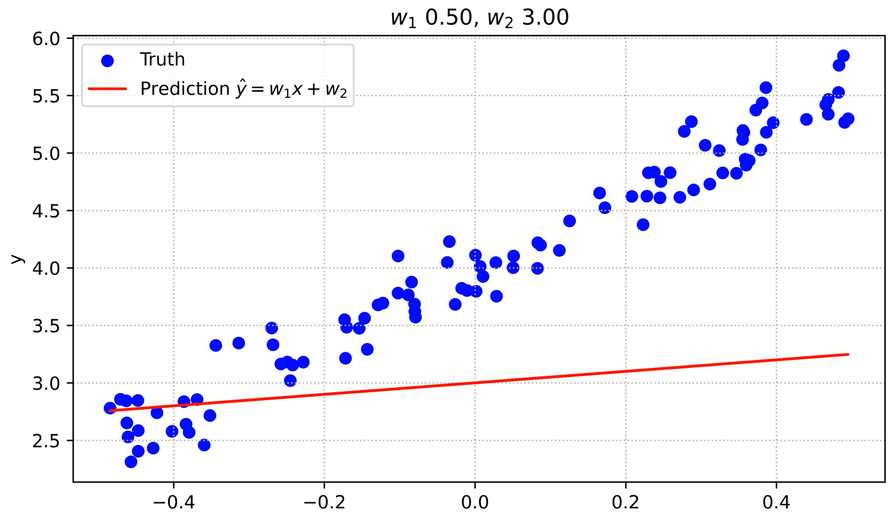
котик

собачка

Обучающая выборка

Архитектура нейросети



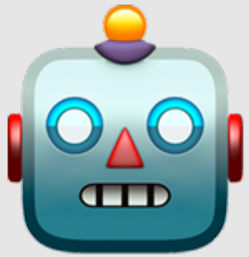


$$W_{k+1} = W_k - \alpha \nabla_W L(W_k)$$

Изменение параметров таким образом, чтобы уменьшать значение функции потерь на обучающей выборке

https://colab.research.google.com/github/MerkulovDaniil/optim/blob/master/assets/Notebooks/GD_2d_visualization.ipynb

СБЕР УНИВЕРСИТЕТ



ПРАКТИКА

Нехватка памяти для обучения больших моделей

```
RuntimeError: cuda runtime error (2) : out of memory at /data/users/soumith/miniconda2/cond
```

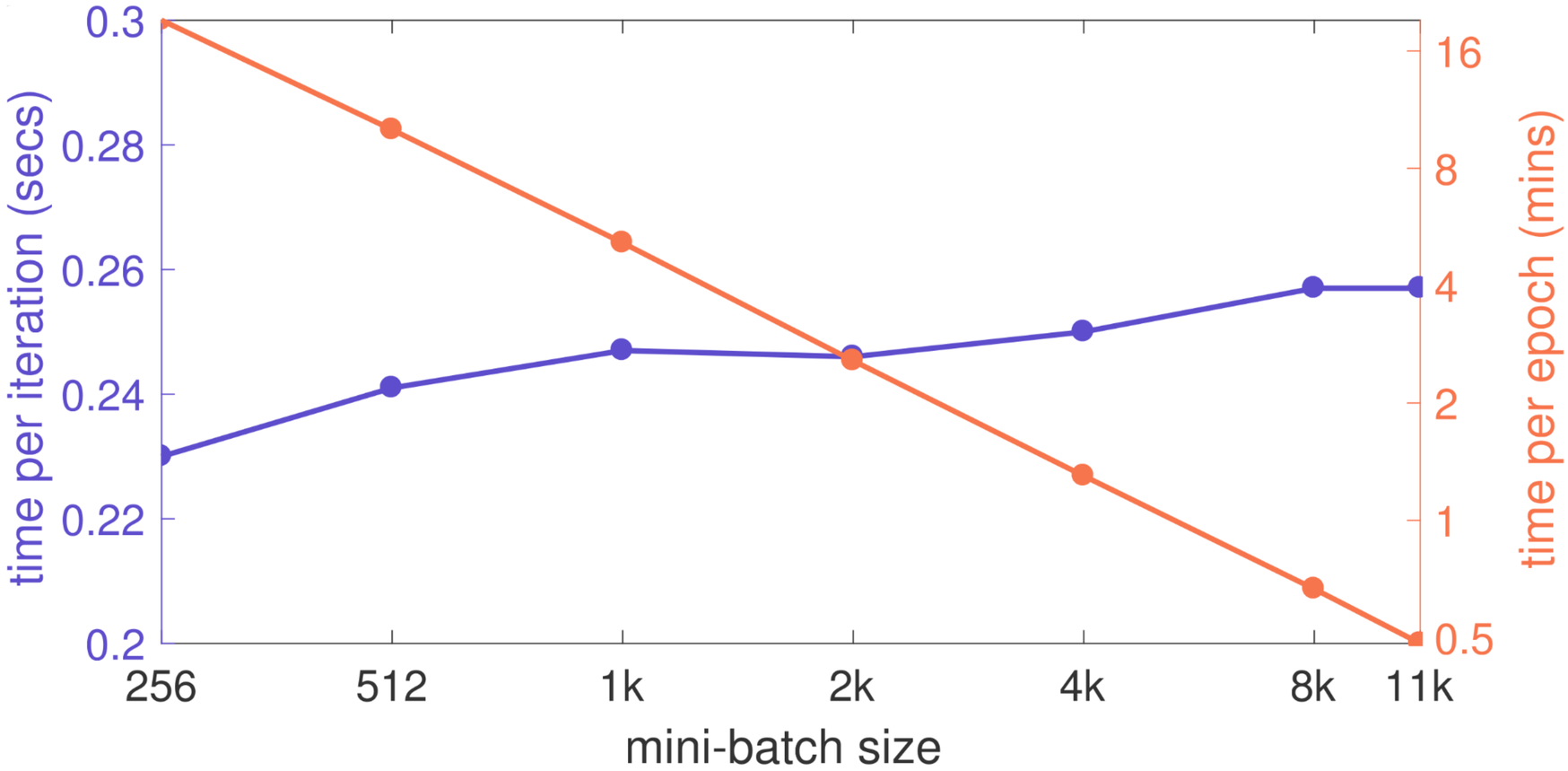
how can i solve this error?

apaszke commented on Mar 8, 2017 — Member

You're running out of memory on the GPU. It's not a bug.

16 3

Размер батча и время, затрачиваемое на одну эпоху



При наличии достаточной памяти GPU, увеличение размера батча позволяет утилизировать ресурсы параллельных вычислений.

📄 **Paper**
Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour.

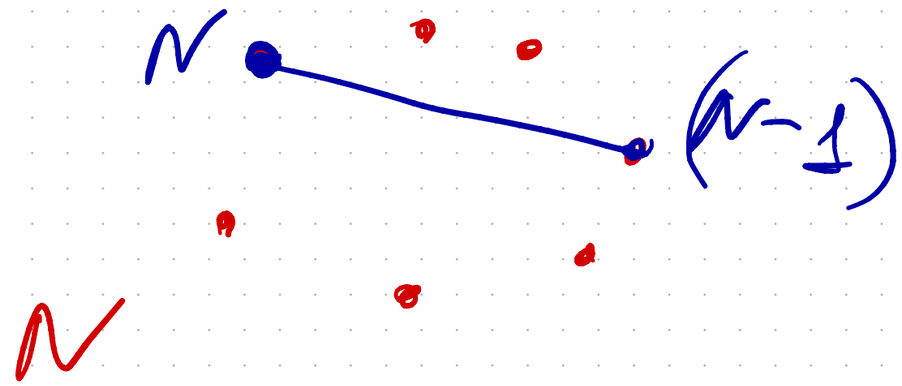
Просто так увеличить размер батча не получится

kn	η	top-1 error (%)
256	0.05	23.92 ±0.10
256	0.10	23.60 ±0.12
256	0.20	23.68 ±0.09
8k	$0.05 \cdot 32$	24.27 ±0.08
8k	$0.10 \cdot 32$	23.74 ±0.09
8k	$0.20 \cdot 32$	24.05 ±0.18
8k	0.10	41.67 ±0.10
8k	$0.10 \cdot \sqrt{32}$	26.22 ±0.03

(a) **Comparison of learning rate scaling rules.** A reference learning rate of $\eta = 0.1$ works best for $kn = 256$ (23.68% error). The linear scaling rule suggests $\eta = 0.1 \cdot 32$ when $kn = 8$k, which again gives best performance (23.74% error). Other ways of scaling η give worse results.

Обучение ResNet-50 на датасете ImageNet с разными вариантами увеличения размера батча.

📄 **Paper**
Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour.

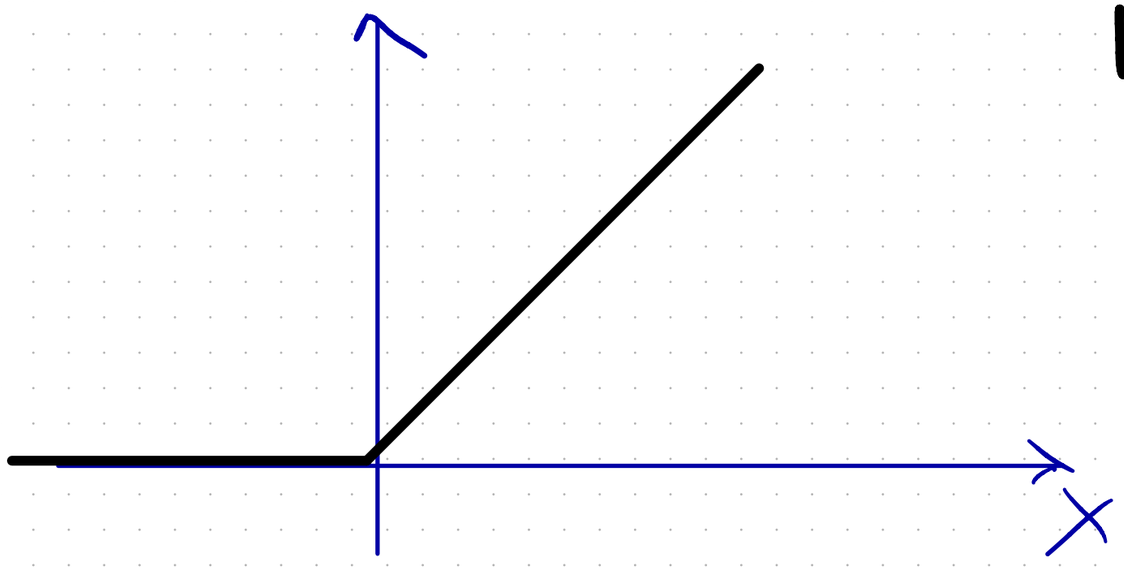


$\frac{N \cdot (N-1)}{2}$ способов

10^{10} нейронов

$\frac{10^{10} \cdot 10^{10}}{2} \approx 10^{20}$ связей

ReLU
Rectified Linear Unit



не линейная

100 слоёв

Выход

$\sigma_{100}(W_{100} \cdot \sigma_{99}(W_{99} \cdot \ldots \cdot \sigma_1(W_1 X)) \ldots)$

$W_{100} \cdot W_{99} \cdot \ldots \cdot W_1 \cdot X$

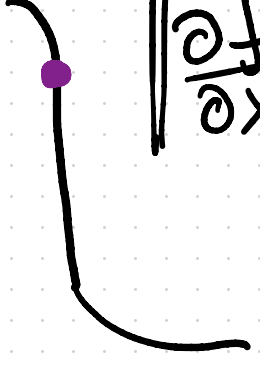
- взрыв / затухание градиентов

$$x_{k+1} = x_k - \left[\nabla^2 f(x_k)\right]^{-1} \cdot \nabla f(x_k)$$

Пусть $\nabla^2 f(x_k) = \mathrm{diag}(m_k) = \begin{pmatrix} m_k^1 & & 0 \\ & \ddots & \\ 0 & & m_k^n \end{pmatrix}$

$$\left[\nabla^2 f(x_k)\right]^{-1} = \mathrm{diag}\left(\frac{1}{m_k}\right)$$

$\left\|\frac{\partial f}{\partial x_p}\right\|$ большая



$\left\|\frac{\partial f}{\partial x_p}\right\|$ маленькая

① Для прототипирования имеет смысл выбирать любой адаптивный алгоритм (AdamW, Nadam, Adam, Rmsprop...) с дефолтными гиперпараметрами.

② Тем не менее, часто хорошо настроенный SGD + momentum показывает лучшие результаты по сравнению с тюненным Adam / AdamW.

Проекция функции потерь нейронной сети

$$L(w) \longrightarrow \min_{w \in \mathbb{R}^p}$$

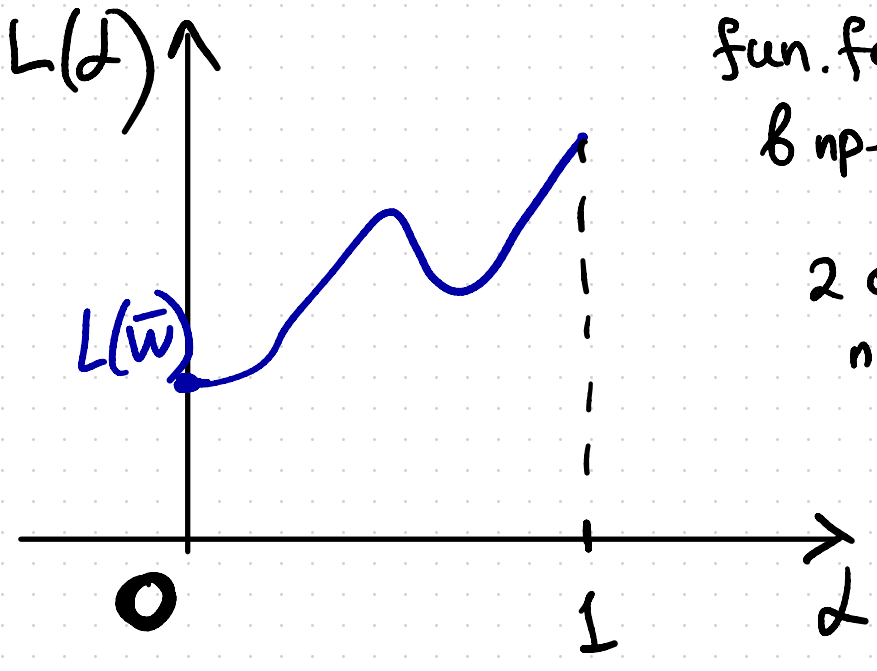
$\overline{w}$ получился после 10 эпох SGD + momentum.

$$w(\alpha) = \overline{w} + \alpha \cdot w_1$$

можно посчитать $w(\alpha)$ для $\alpha = np.linspace(0, 1)$

w_1 - случ. вектор $\in \mathbb{R}^p$

$L(w(\alpha))$



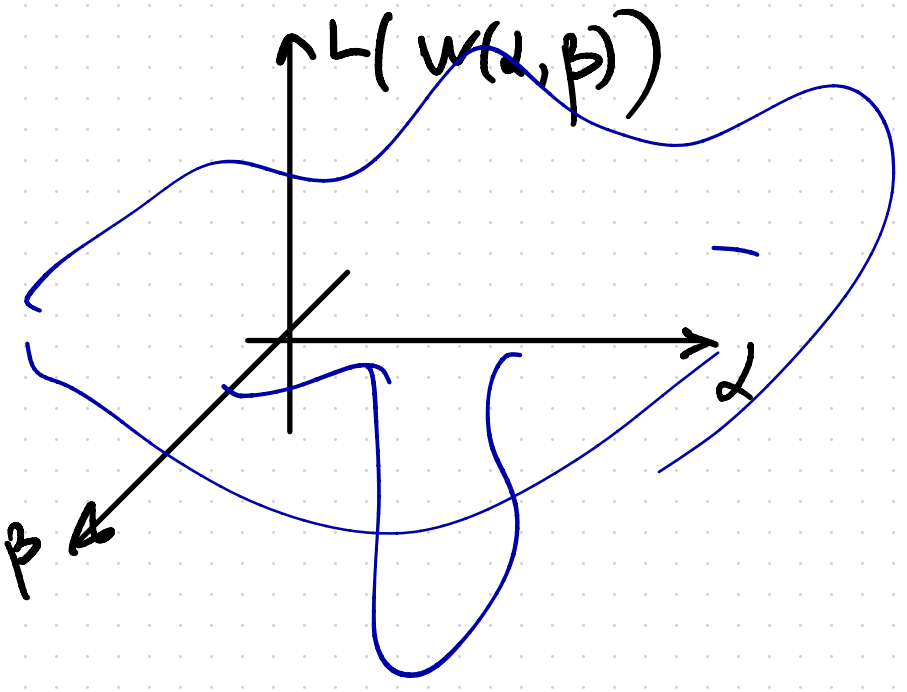
fun. fact

в пр-ве большой размерности

2 случ. вектора почти ортогональны

$$w(\alpha, \beta) = \bar{w} + \alpha \cdot w_1 + \beta \cdot w_2$$

w_1, w_2 – случ. вектора



Generalization error

SHARPNESS AWARE MINIMIZATION (SAM)

если всё пр-во

1 → 5000 · 6 z_1

2 → 4000 · 6 z_2

3 → 2000 · 6 z_3

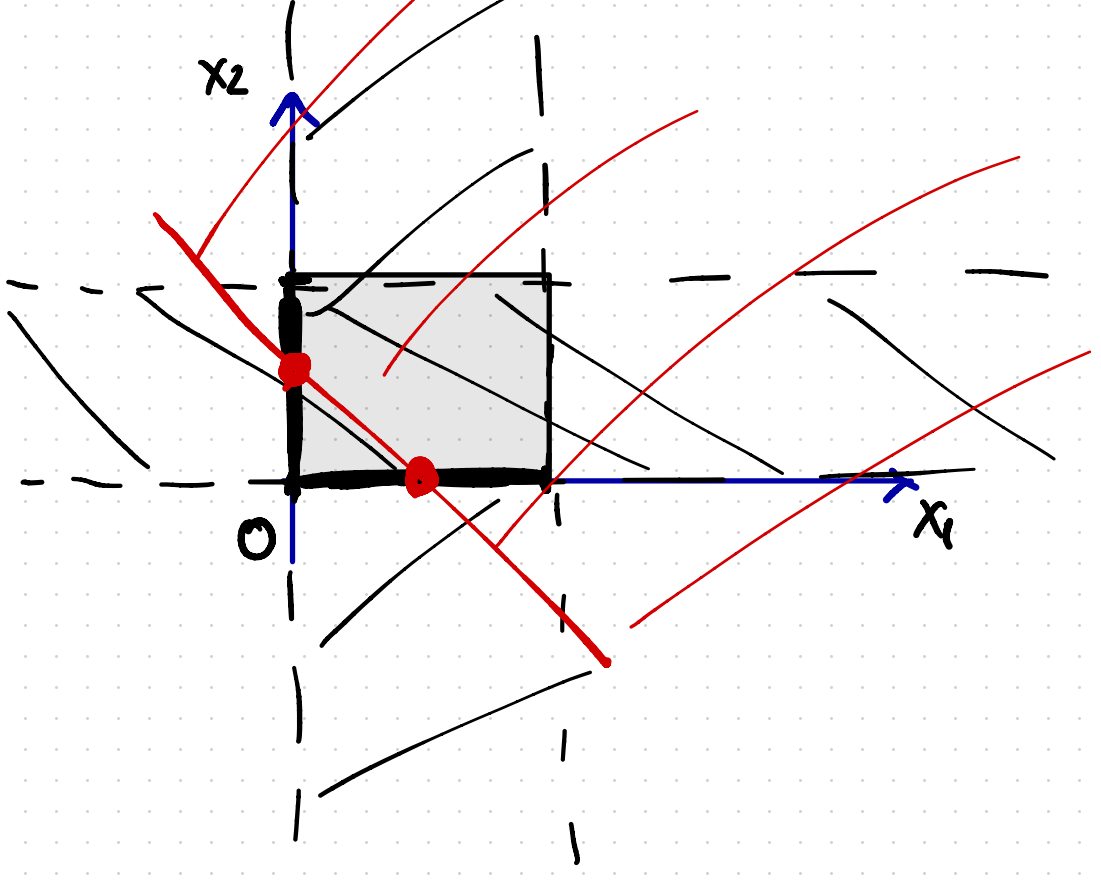
Если фабрика за неделю производит:

$$y_1 + y_2 + y_3 \leq \frac{z_1 + z_2 + z_3}{3}$$

фабрика производит в неделю:

$$\frac{y_1}{z_1} + \frac{y_2}{z_2} + \frac{y_3}{z_3} \leq 1$$

capacity


x2
0
x1