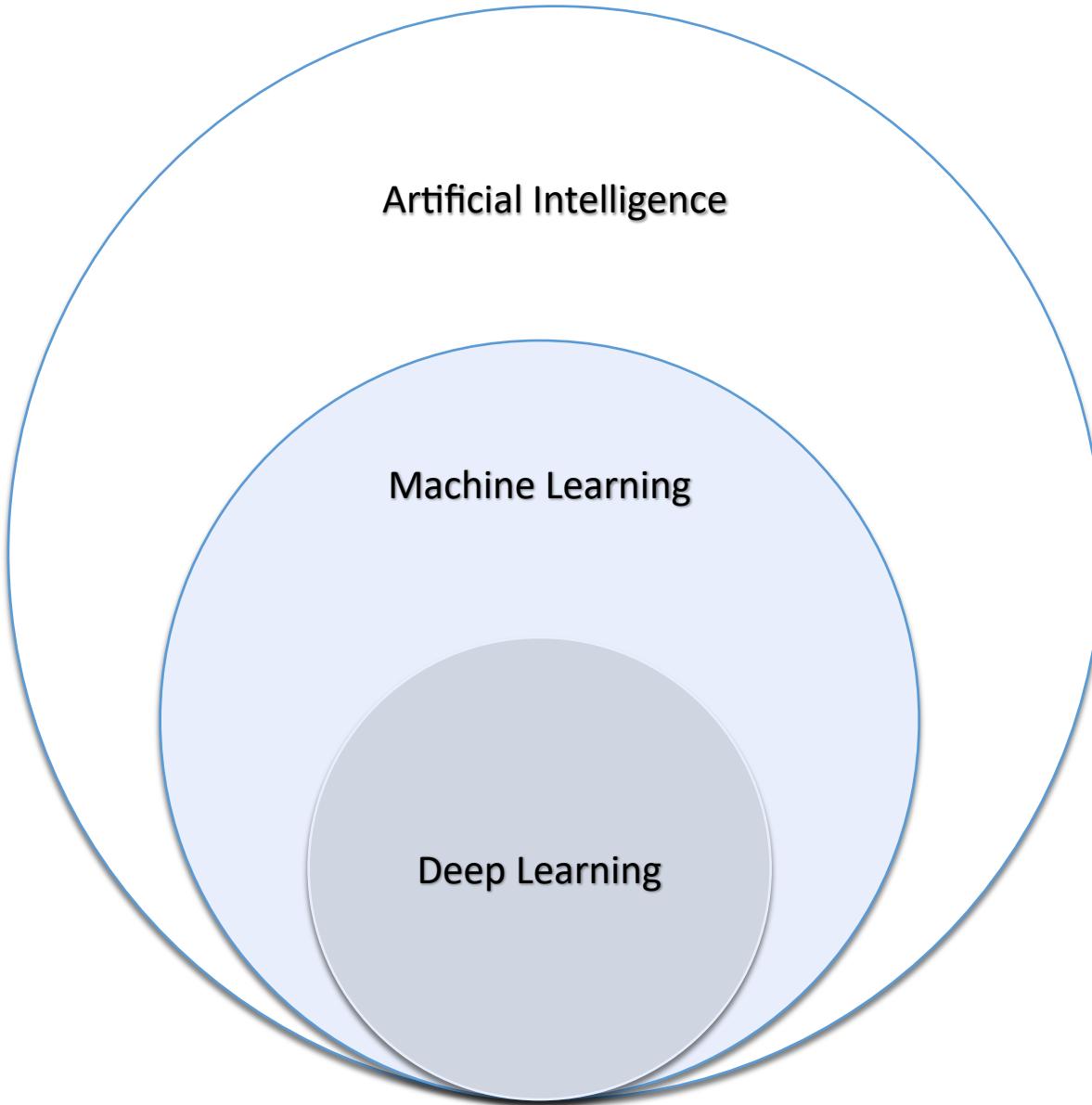


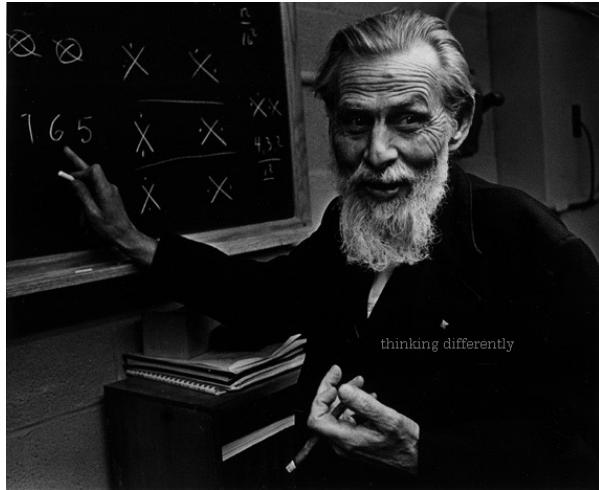
Основы нейронных сетей

Занятие 10

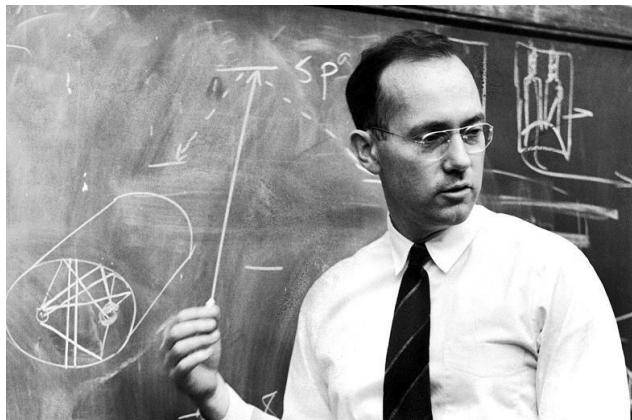


Краткая история ИИ

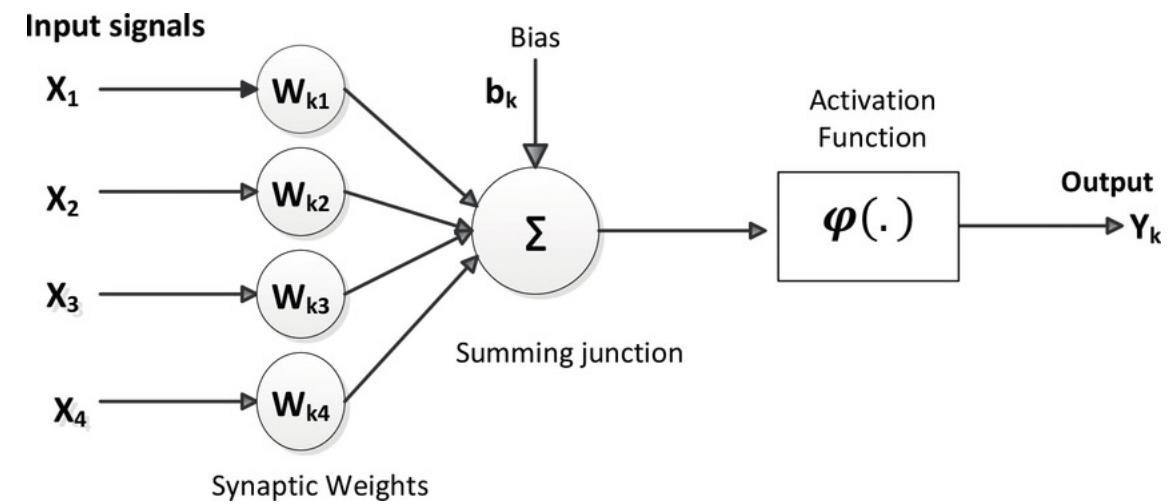
1943: первый формальный нейрон



Уоррен Маккалок



Уолтер Питтс



1956: семинар летом в Дартмунде

Джон МакКарти, Марвин Минский, Клод Шеннон и Натаниэль Рочестер

*"We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. **We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.**"*

Зима близко

- 1956 — Дортмундский семинар, море оптимизма
- 1958 — Персепtron Розенблатта
- середина 1960-х — провал крупного проекта по машинному переводу с русского на английский и наоборот
- 1969 — Марвин Минский и Сеймур Пейперт опубликовали книгу «Персептроны» с критикой

Зима наступила

- Зима искусственного интеллекта — период в истории исследований искусственного интеллекта, связанный с сокращением финансирования и снижением интереса
- Две длительные «зимы» относят к периодам 1974—1980 годов и 1987—1993 годов
- Несмотря на спад финансирования, исследования продолжались

Оттепель

- 1970-е — Расцвет экспертных систем, принимающих решения на основе большого числа правил и знаний о предметной области. MYCIN накопила около 600 правил для идентификации вирусных бактерий и выдачи подходящего метода лечения (угадывала в 69% случаев, лучше любого начинающего врача)
- 1980-е — появилось много разных архитектур
- 1980-е — алгоритм обратного распространения ошибки (backpropagation) позволил обучать сети за линейное время
- Ренессанс нейронных сетей

Снова зима

- Новая волна оптимизма
- 1986 — один из первых AI-отделов экономил компании DEC около 10 миллионов долларов в год
- Завышенные ожидания снова лопнули
- 1990-е — ударными темпами развивается классическое машинное обучение

Революция 2000x

- 2005-2006 — группы Хинтона и Бенджи научились обучать глубокие нейросети
- На протяжении 60 лет акцент был на алгоритмах. Более свежие исследования показывают, что важнее иметь данные, нежели беспокоиться об алгоритмах

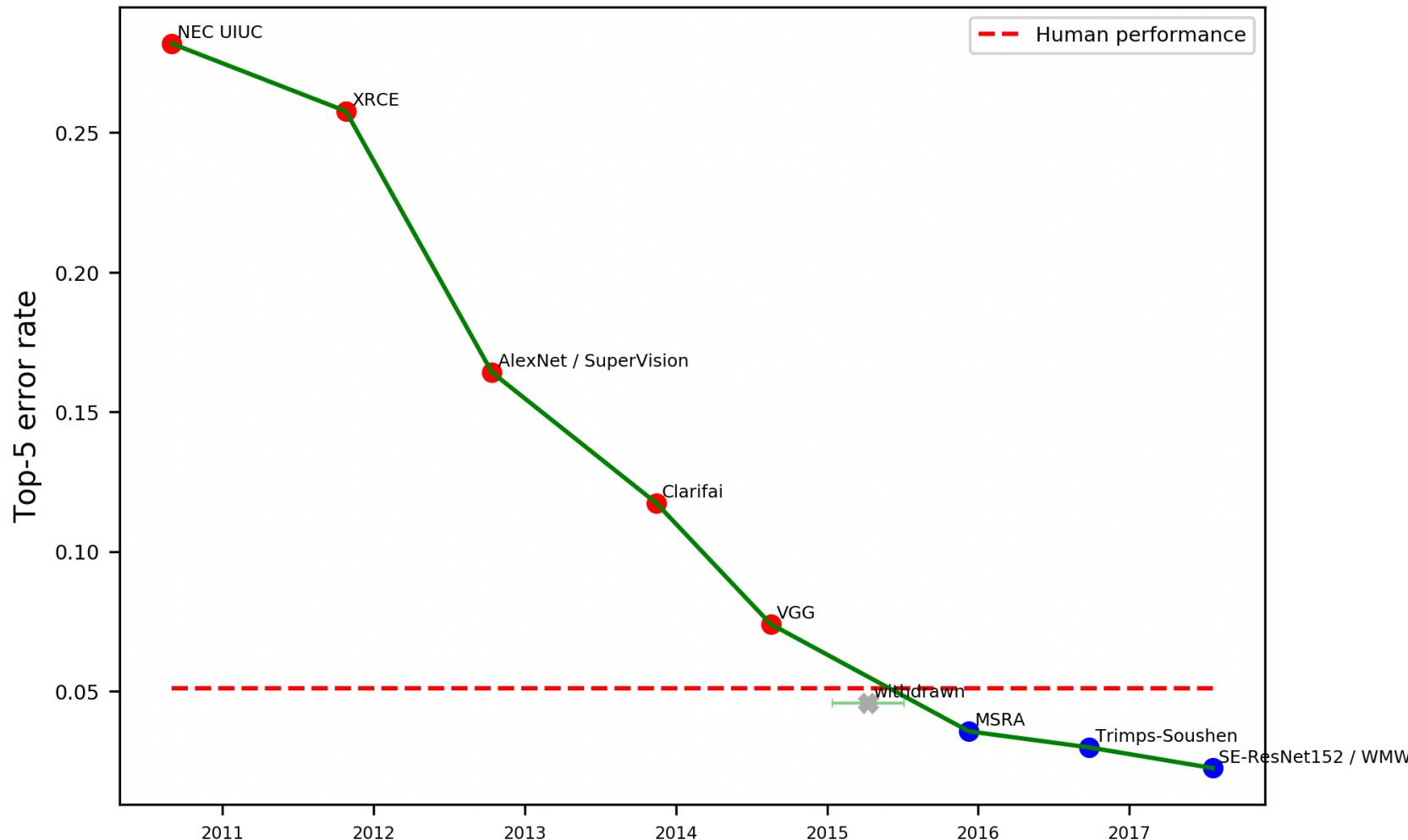
Революция 2000x

Year	Breakthroughs in AI	Datasets (First Available)	Algorithms (First Proposed)
1994	Human-level spontaneous speech recognition	Spoken Wall Street Journal articles and other texts (1991)	Hidden Markov Model (1984)
1997	IBM Deep Blue defeated Garry Kasparov	700,000 Grandmaster chess games, aka “The Extended Book” (1991)	Negascout planning algorithm (1983)
2005	Google’s Arabic- and Chinese-to-English translation	1.8 trillion tokens from Google Web and News pages (collected in 2005)	Statistical machine translation algorithm (1988)
2011	IBM Watson became the world Jeopardy! champion	8.6 million documents from Wikipedia, Wiktionary, Wikiquote, and Project Gutenberg (updated in 2010)	Mixture-of-Experts algorithm (1991)
2014	Google’s GoogLeNet object classification at near-human performance	ImageNet corpus of 1.5 million labeled images and 1,000 object categories (2010)	Convolution neural network algorithm (1989)
2015	Google’s Deepmind achieved human parity in playing 29 Atari games by learning general control from video	Arcade Learning Environment dataset of over 50 Atari games (2013)	Q-learning algorithm (1992)
Average No. of Years to Breakthrough:		3 years	18 years

Важные тренды

#1 Точность растет

Imagenet Image Recognition



#2 Сложность растет

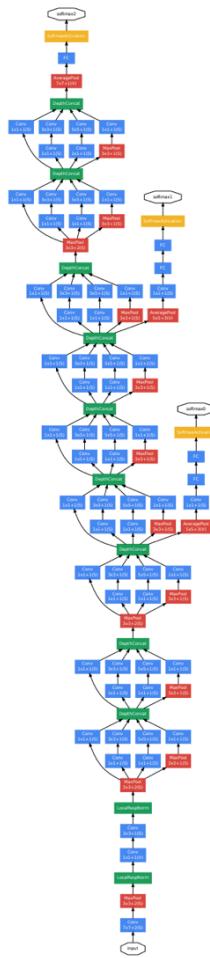
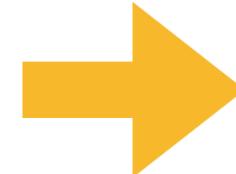
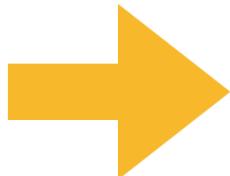
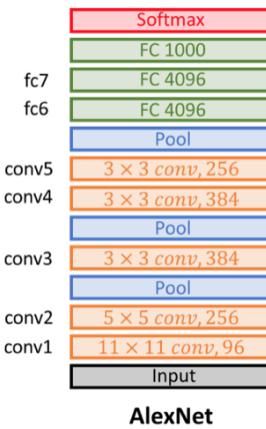
Последний рекордсмен:
Триллион параметров

SWITCH TRANSFORMERS: SCALING TO TRILLION
PARAMETER MODELS WITH SIMPLE AND EFFICIENT
SPARSITY

William Fedus*
Google Brain
liamfedus@google.com

Barret Zoph*
Google Brain
barrettzoph@google.com

Noam Shazeer
Google Brain
noam@google.com



#3 Объем данных растет



4,802,352,891

Internet Users in the world



1,830,708,336

Total number of Websites



150,464,088,337

Emails sent **today**



4,308,108,712

Google searches **today**



4,146,395

Blog posts written **today**



450,806,077

Tweets sent **today**



4,251,228,756

Videos viewed **today**
on YouTube



50,605,789

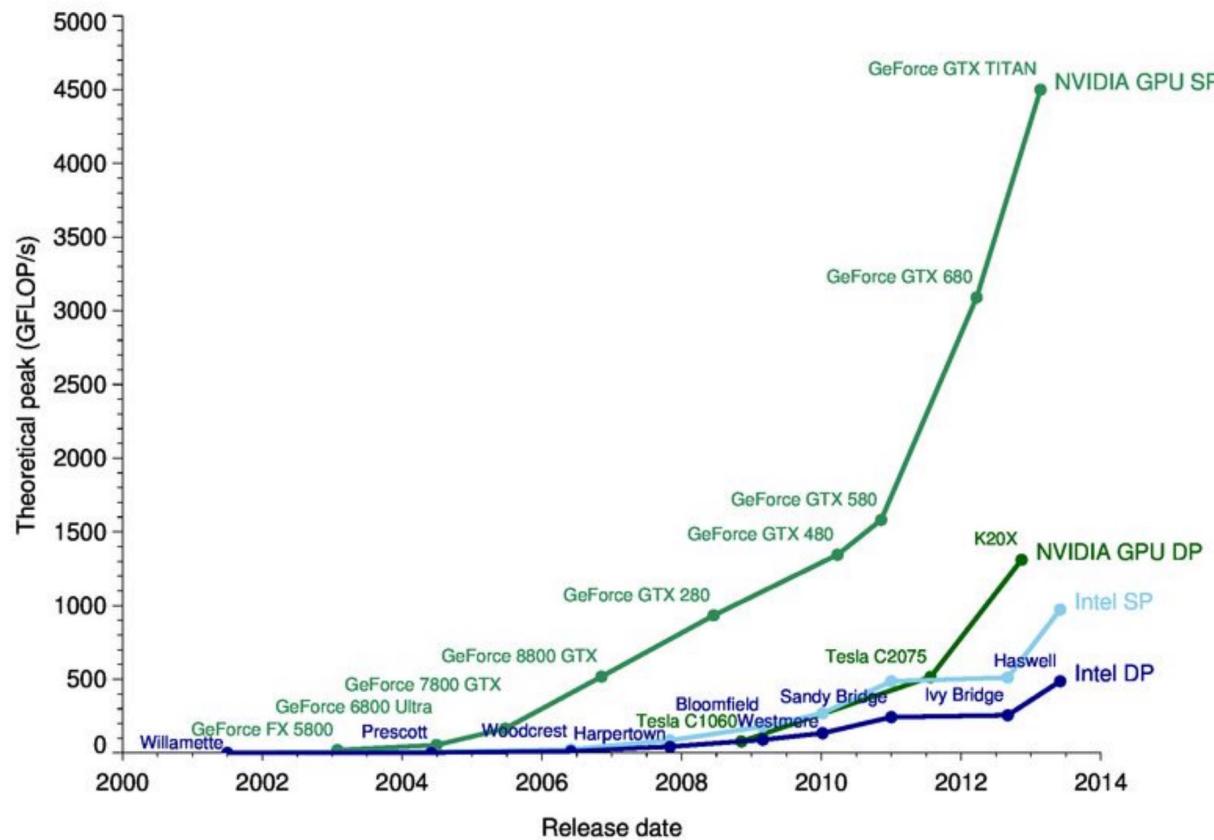
Photos uploaded **today**
on Instagram



88,339,242

Tumblr posts **today**

#4 Вычислительные мощности растут



Свежие примеры

Генерация картинок по тексту

TEXT PROMPT

an armchair in the shape of an avocado [...]

AI-GENERATED IMAGES

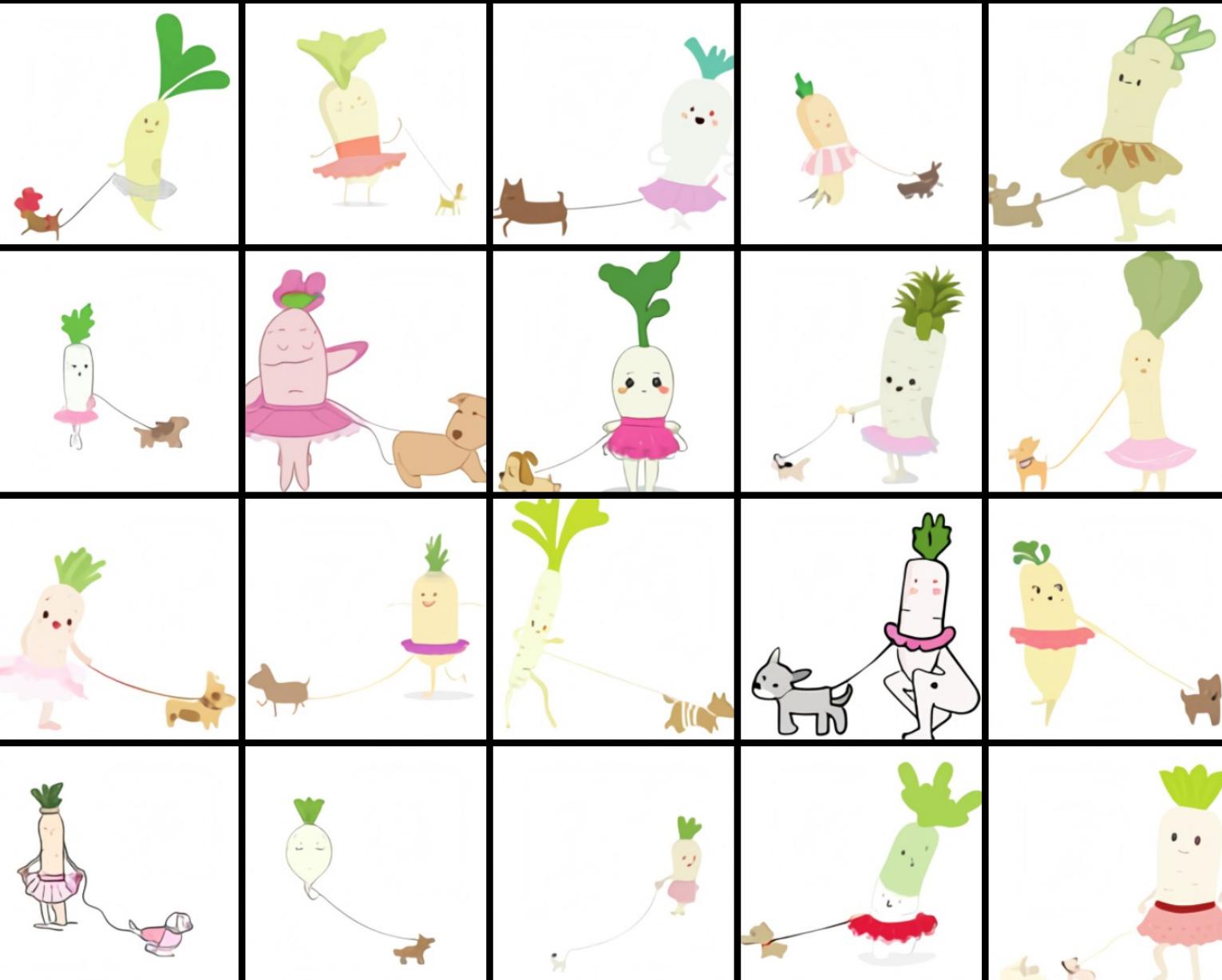


DALL-E is a 12-billion parameter version of [GPT-3](#) trained to generate images from text descriptions, using a dataset of text–image pairs. We've found that it has a diverse set of capabilities, including creating anthropomorphized versions of animals and objects, combining unrelated concepts in plausible ways, rendering text, and applying transformations to existing images.

TEXT PROMPT

an illustration of a baby daikon radish in a tutu walking a dog

AI-GENERATED IMAGES



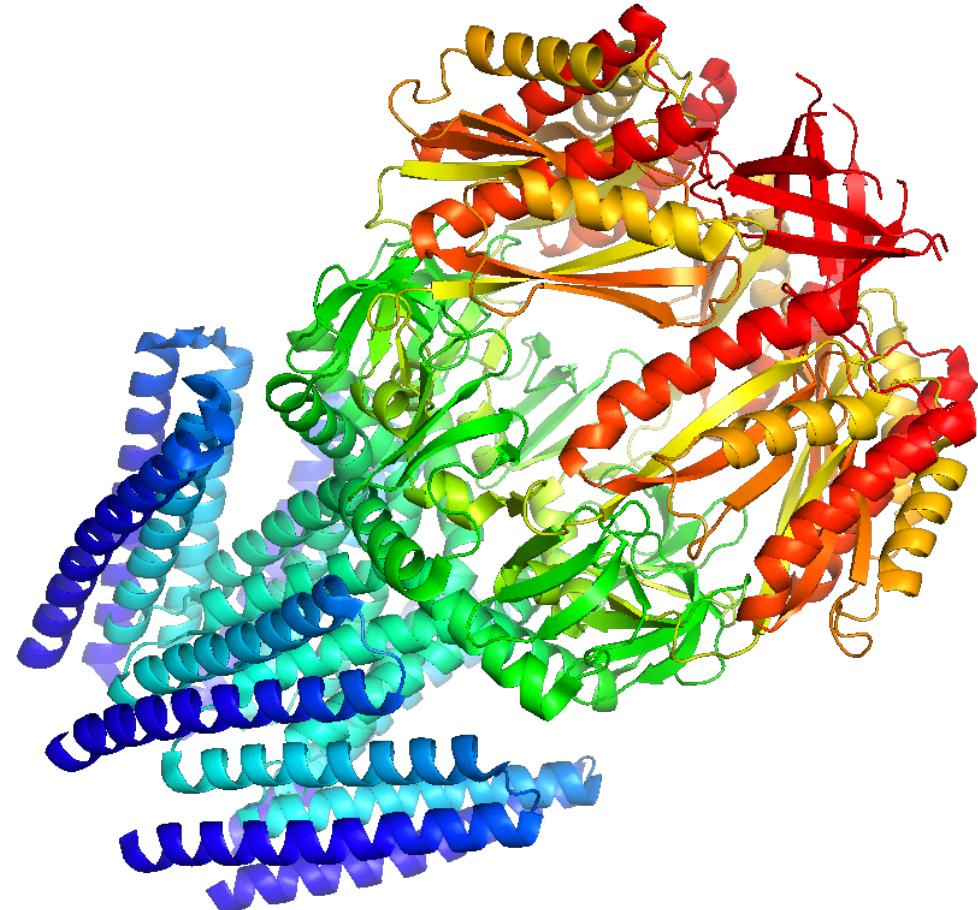
Перенос стиля



<https://colab.research.google.com/github/GeorgeDavila/Renify/blob/main/RenifyColab.ipynb#scrollTo=qI5EUBPBuAji>

Предсказание структуры белка

Here we show that we can train a neural network to make accurate predictions of the distances between pairs of residues, which convey more information about the structure than contact predictions. Using this information, we construct a potential of mean force⁴ that can accurately describe the shape of a protein. We find that the resulting potential can be optimized by a simple gradient descent algorithm to generate structures without complex sampling procedures



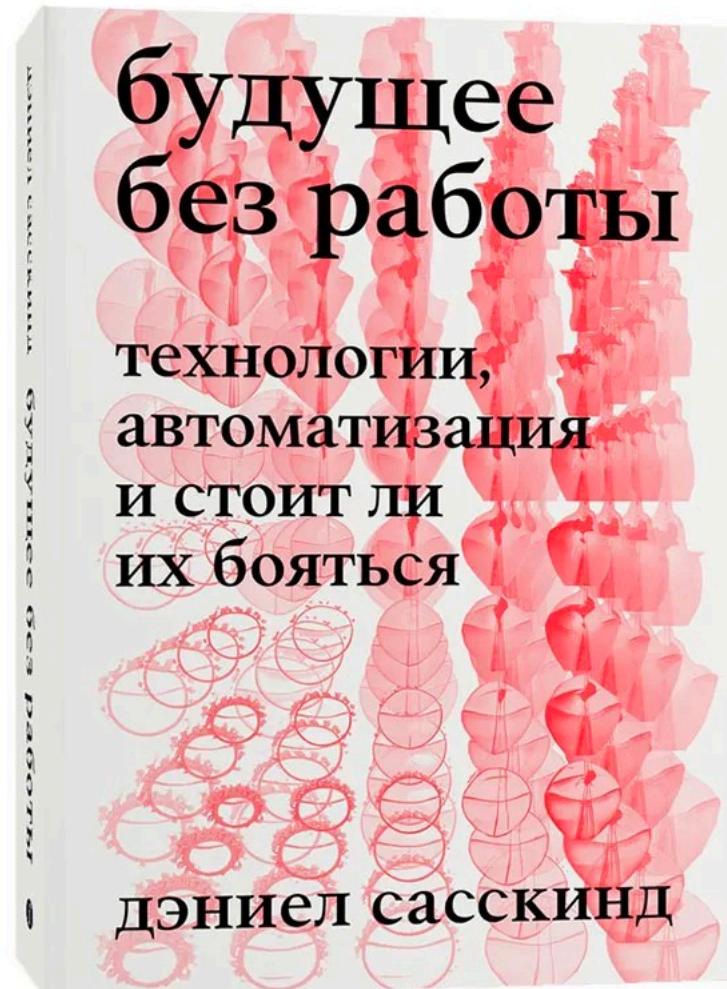
Детекция объектов

Scaled YOLO v4 является самой лучшей нейронной сетью для обнаружения объектов — самой точной нейронной сетью (55.8% AP) на датасете Microsoft COCO среди всех опубликованных нейронных сетей на данный момент. А также является лучшей с точки зрения соотношения скорости к точности во всем диапазоне точности и скорости от 15 FPS до 1774 FPS. На данный момент это Top1 нейронная сеть для обнаружения объектов.



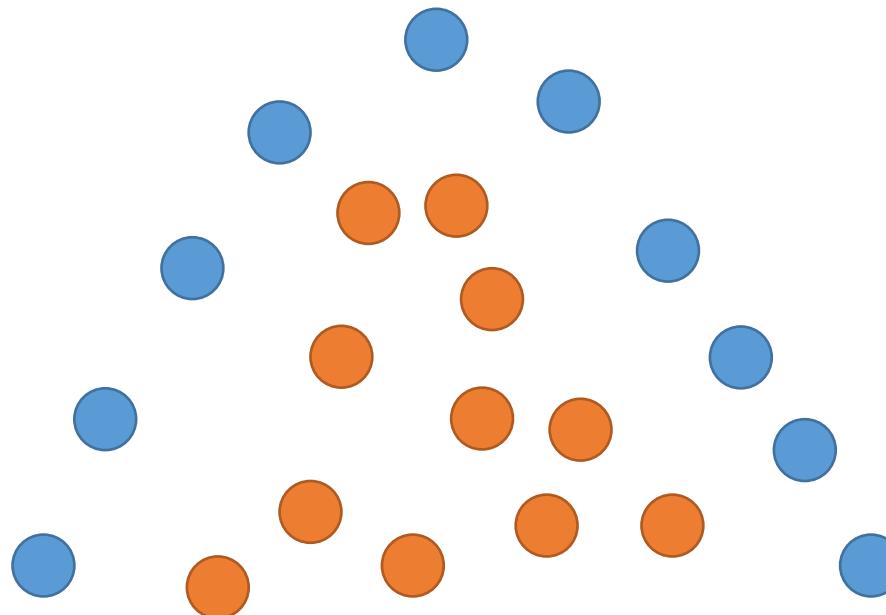
Машинный перевод

За 40 секунд «Яндекс.Переводчик» справился с 350-страничной книгой британца, состоящей из 650 тысяч знаков.

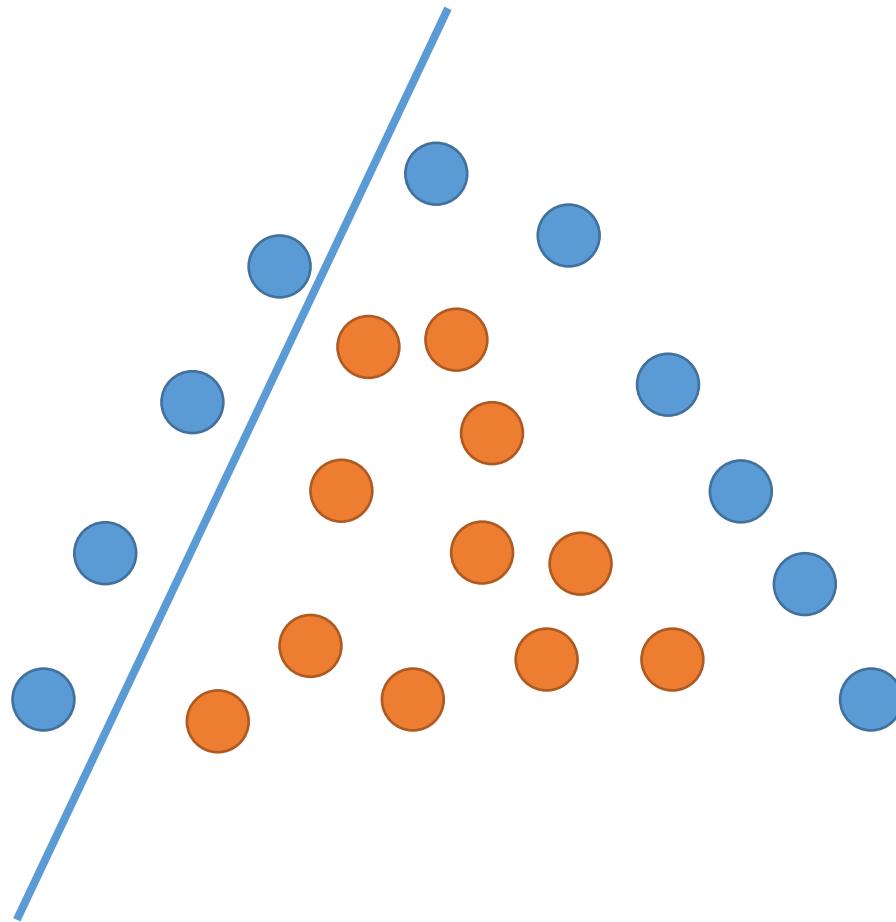


К нейросетям

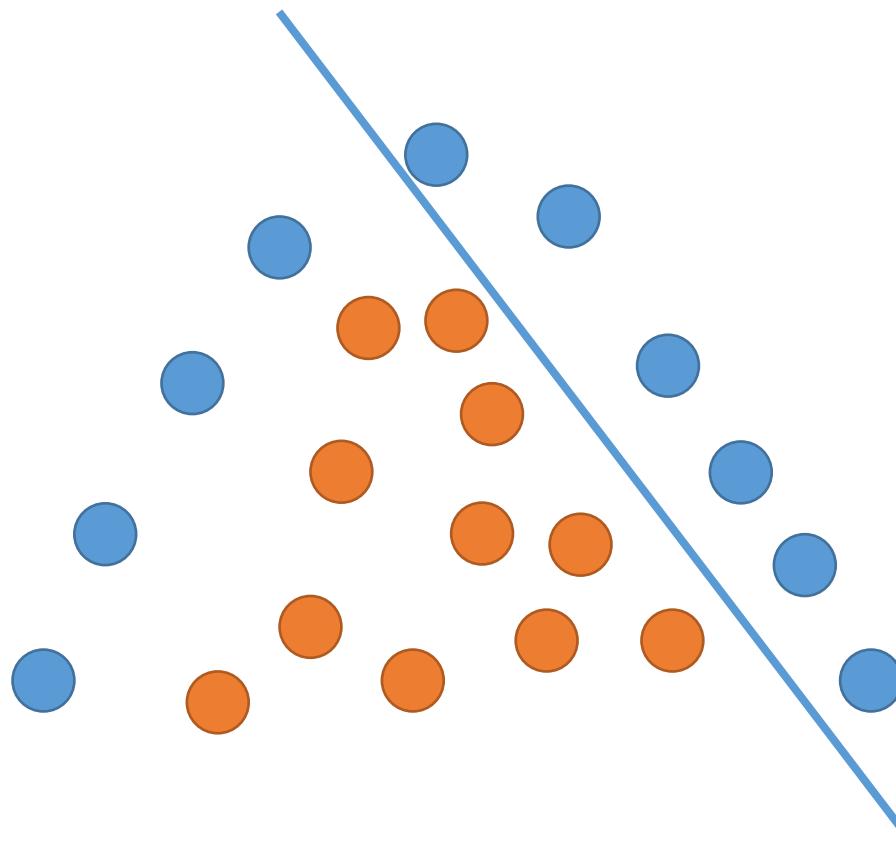
Нелинейные закономерности



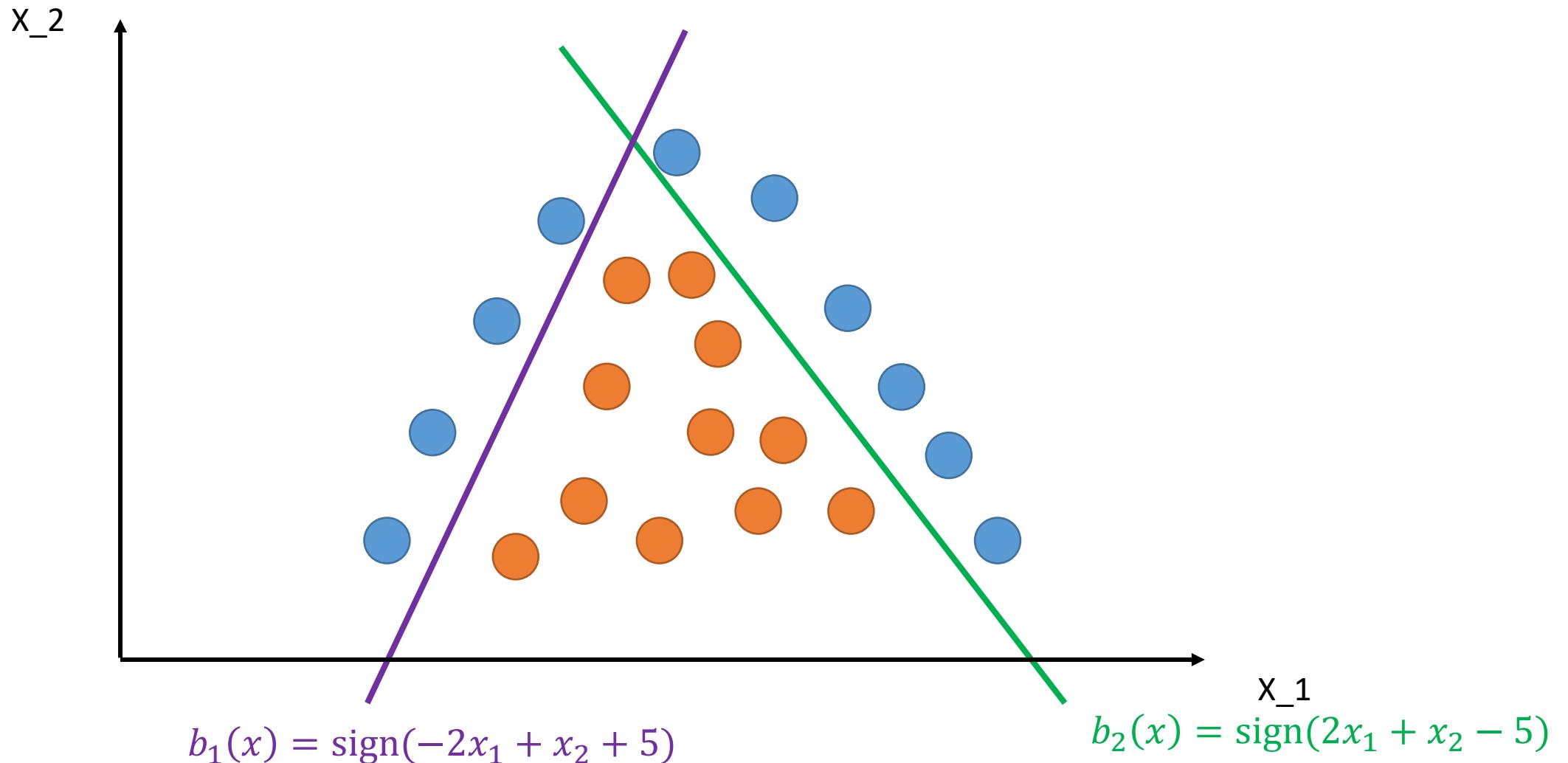
Нелинейные закономерности



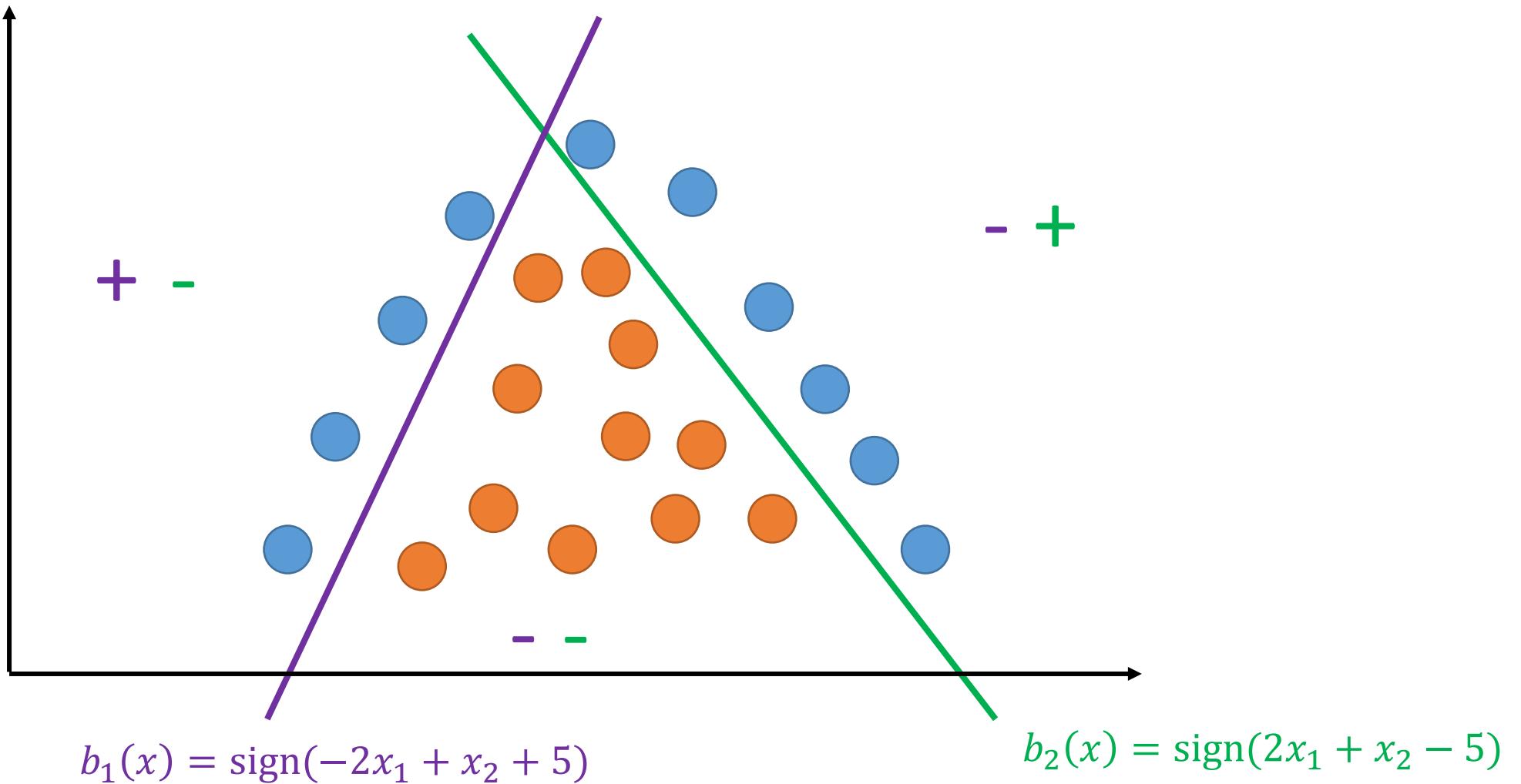
Нелинейные закономерности



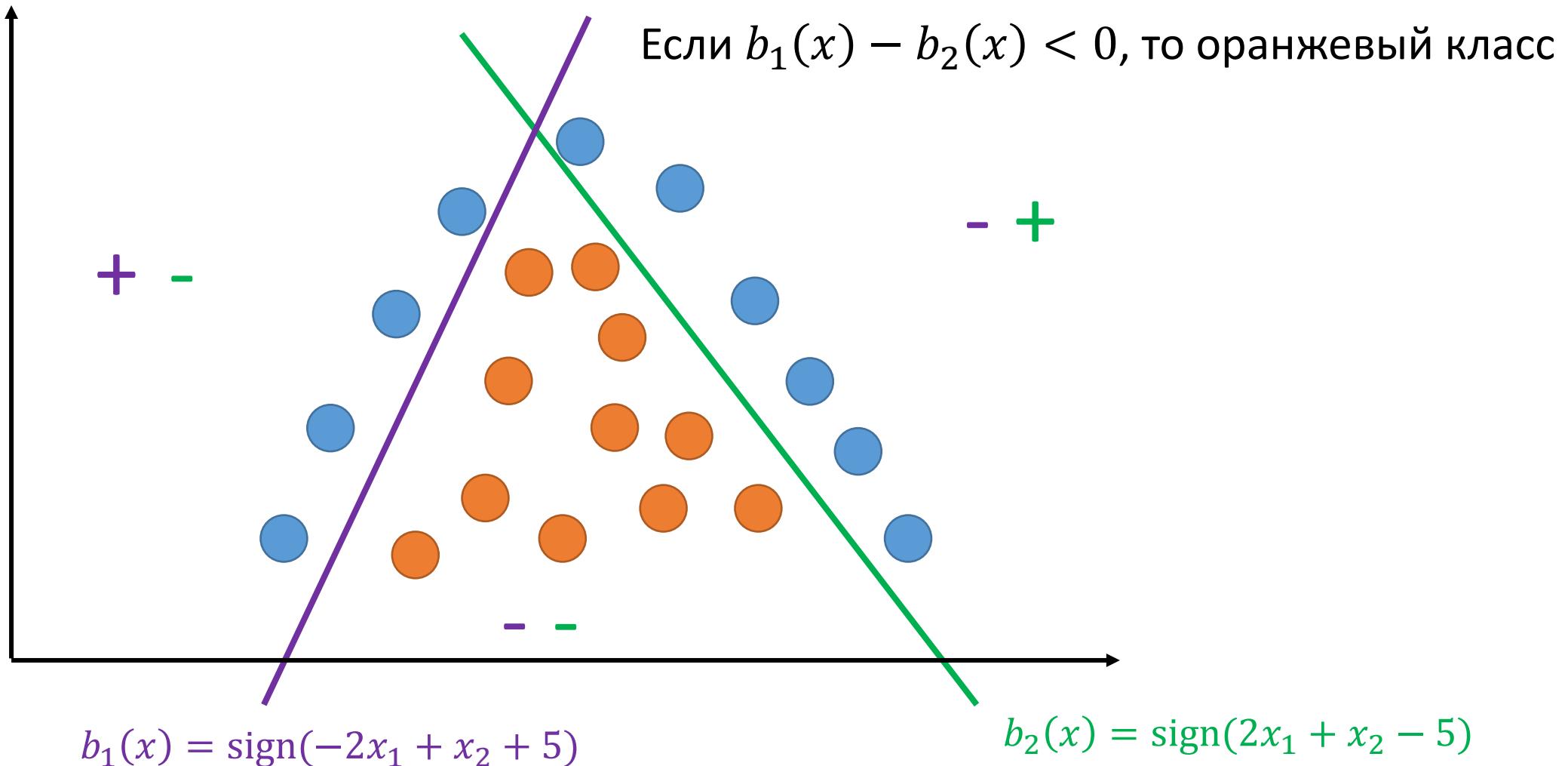
Нелинейные закономерности



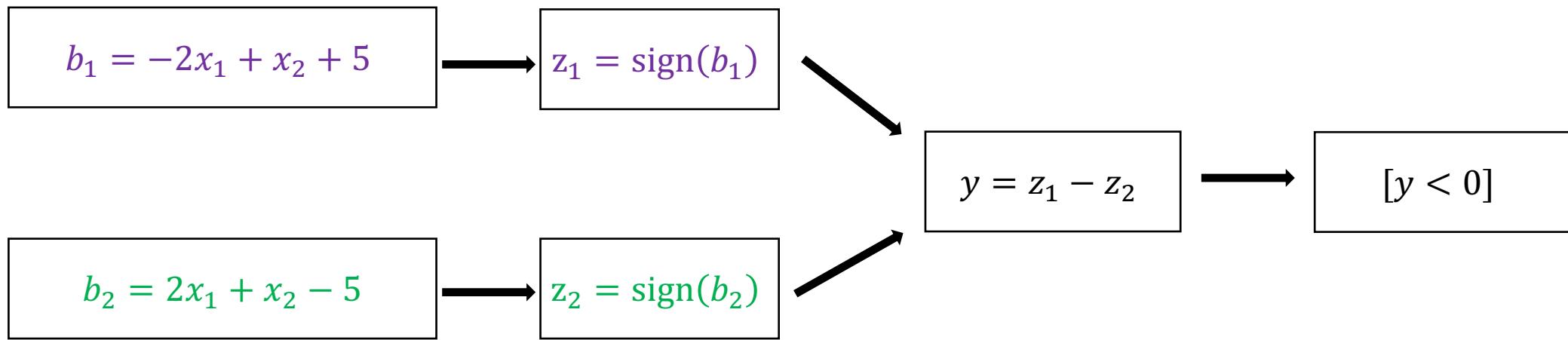
Нелинейные закономерности



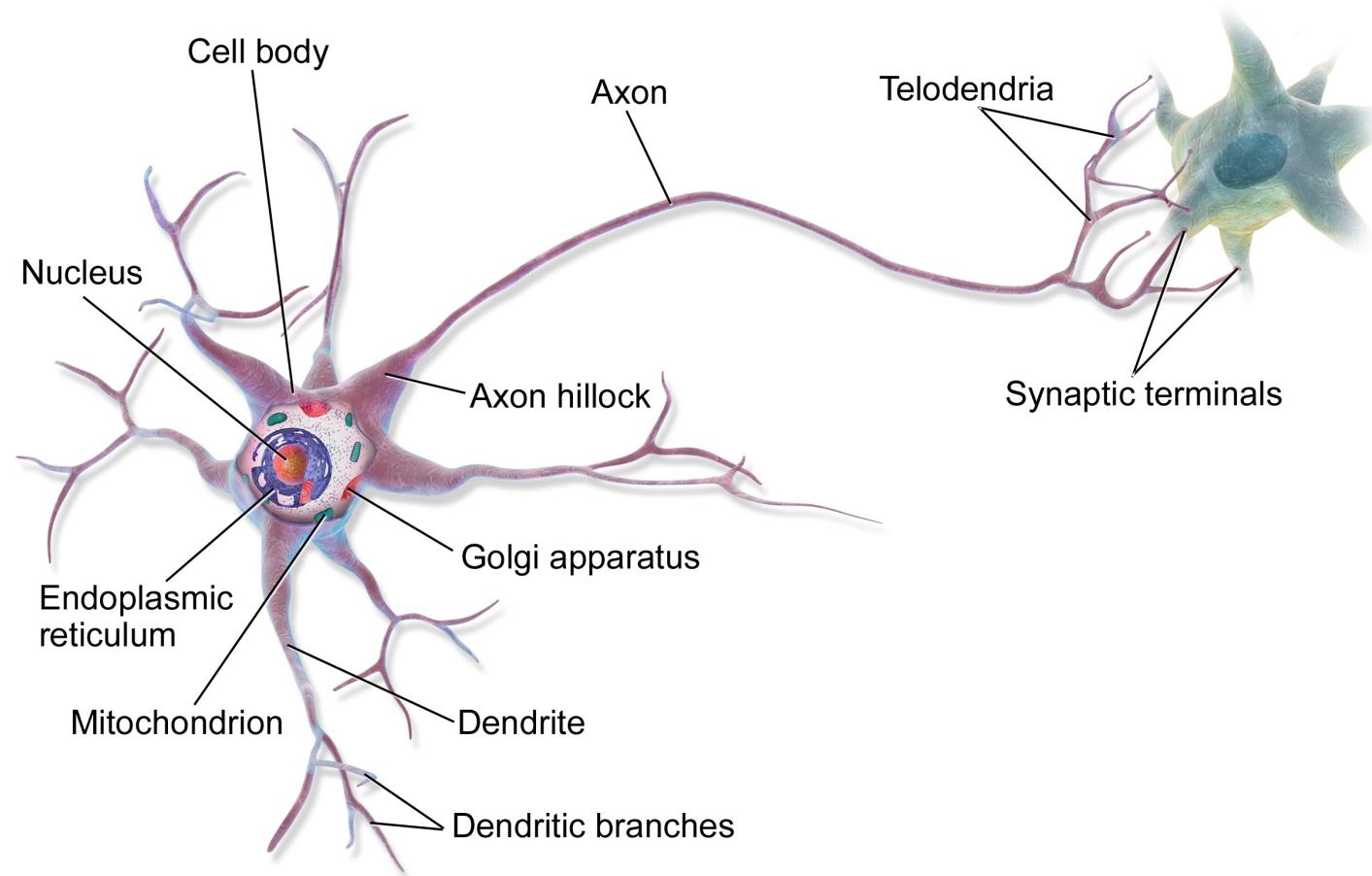
Нелинейные закономерности



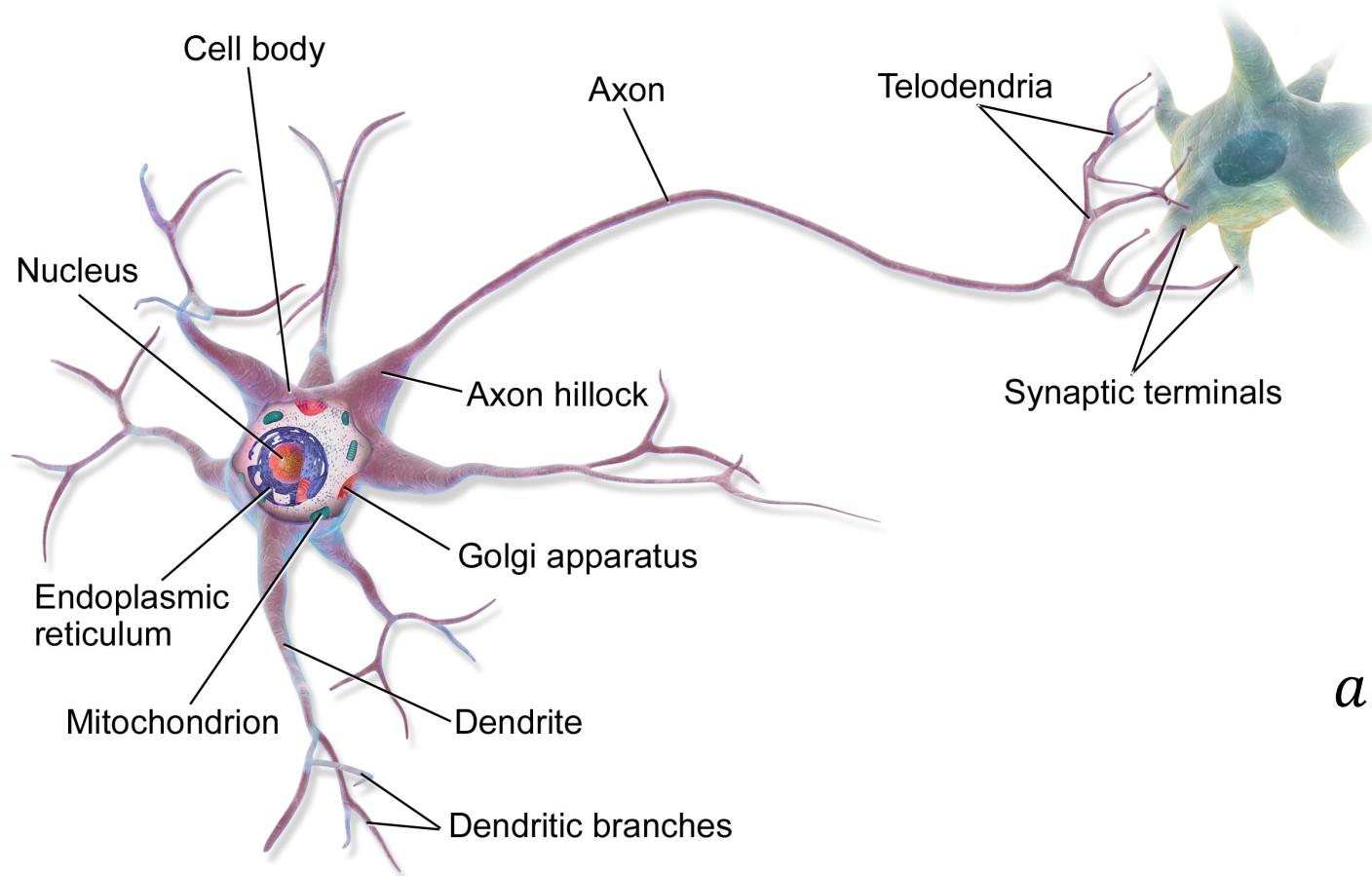
Нелинейные закономерности



Нейрон

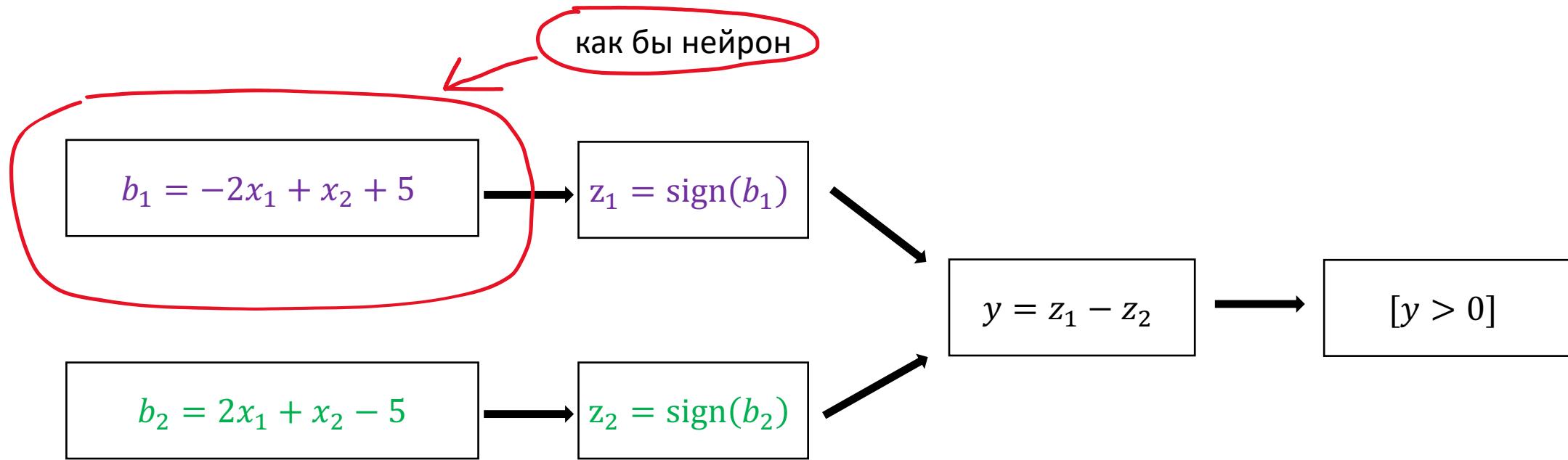


Нейрон

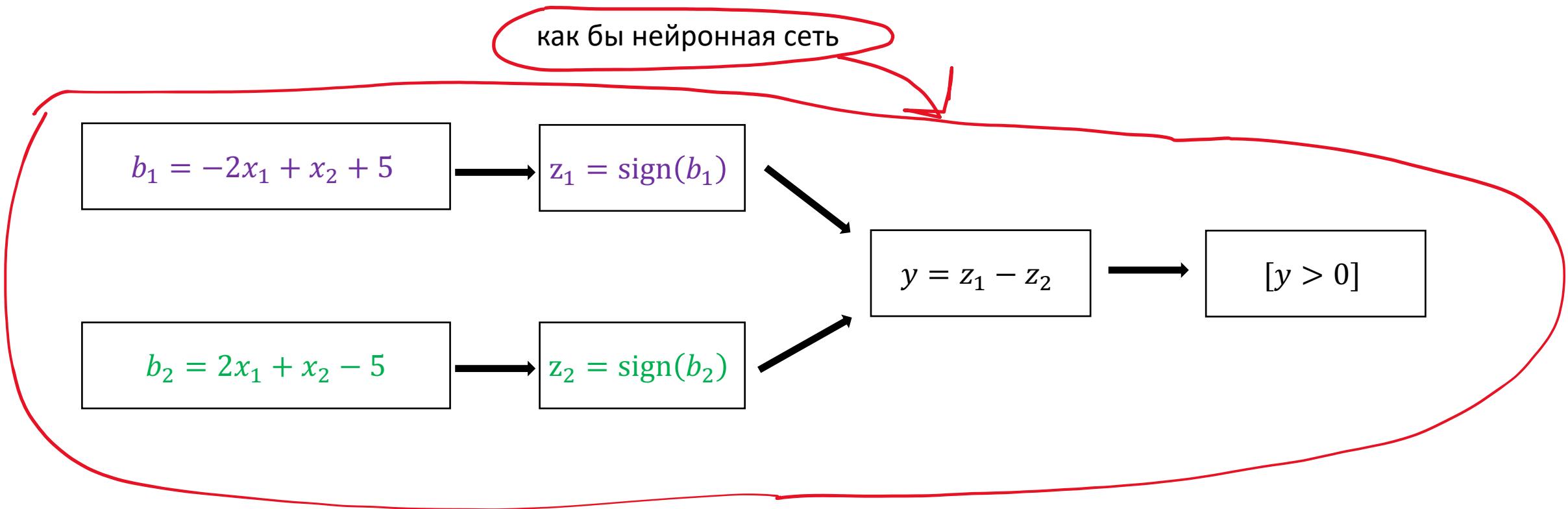


$$a(x) = \sum_{j=1}^d w_j x_j$$

Нелинейные закономерности

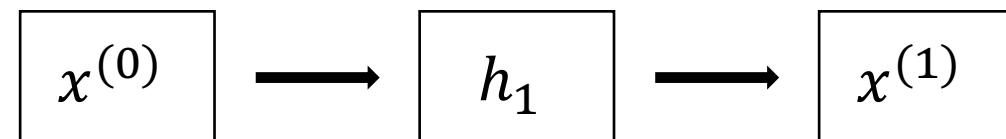


Нелинейные закономерности

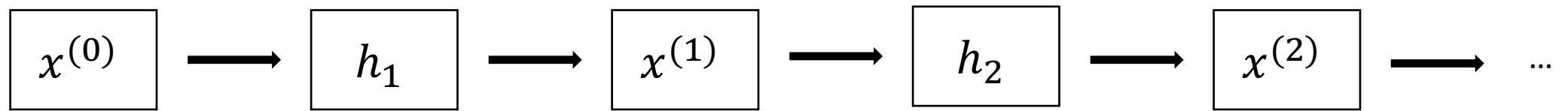


Граф вычислений (или нейронная сеть)

- $x^{(0)}$ — признаки объекта
- $h_1(x)$ — преобразование («слой»)
- $x^{(1)}$ — результат

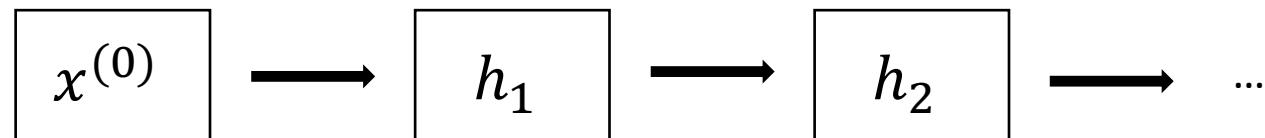
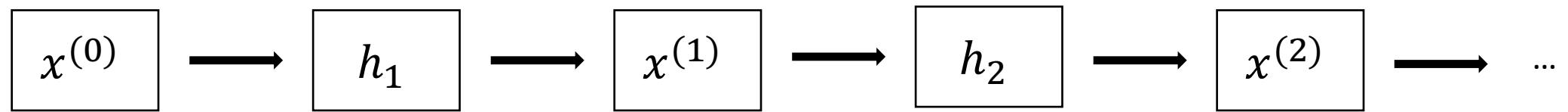


Граф вычислений (или нейронная сеть)

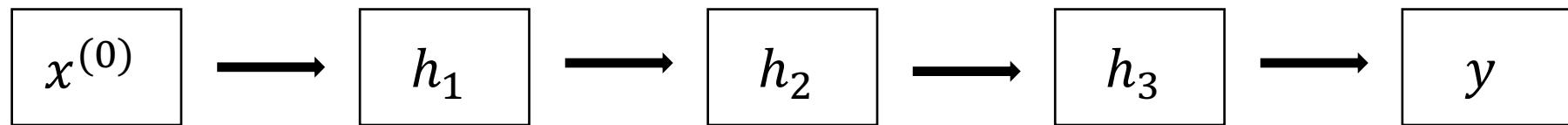


Граф вычислений (или нейронная сеть)

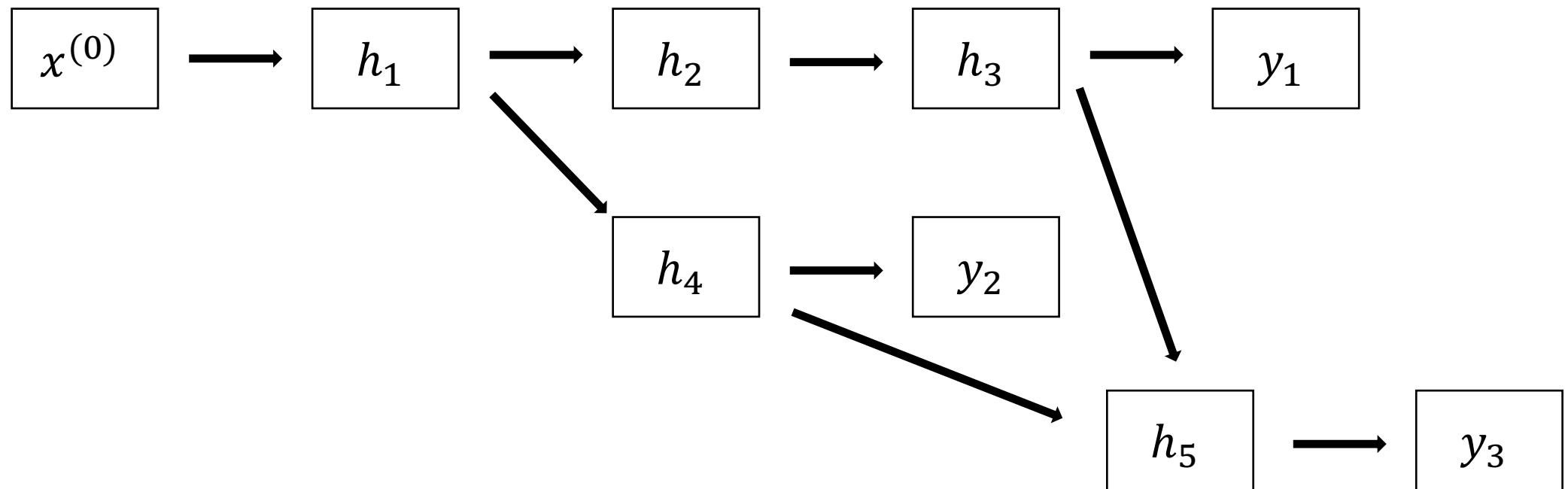
Для простоты не будем рисовать промежуточные результаты



Граф вычислений (или нейронная сеть)



Граф вычислений (или нейронная сеть)



Полносвязные слои

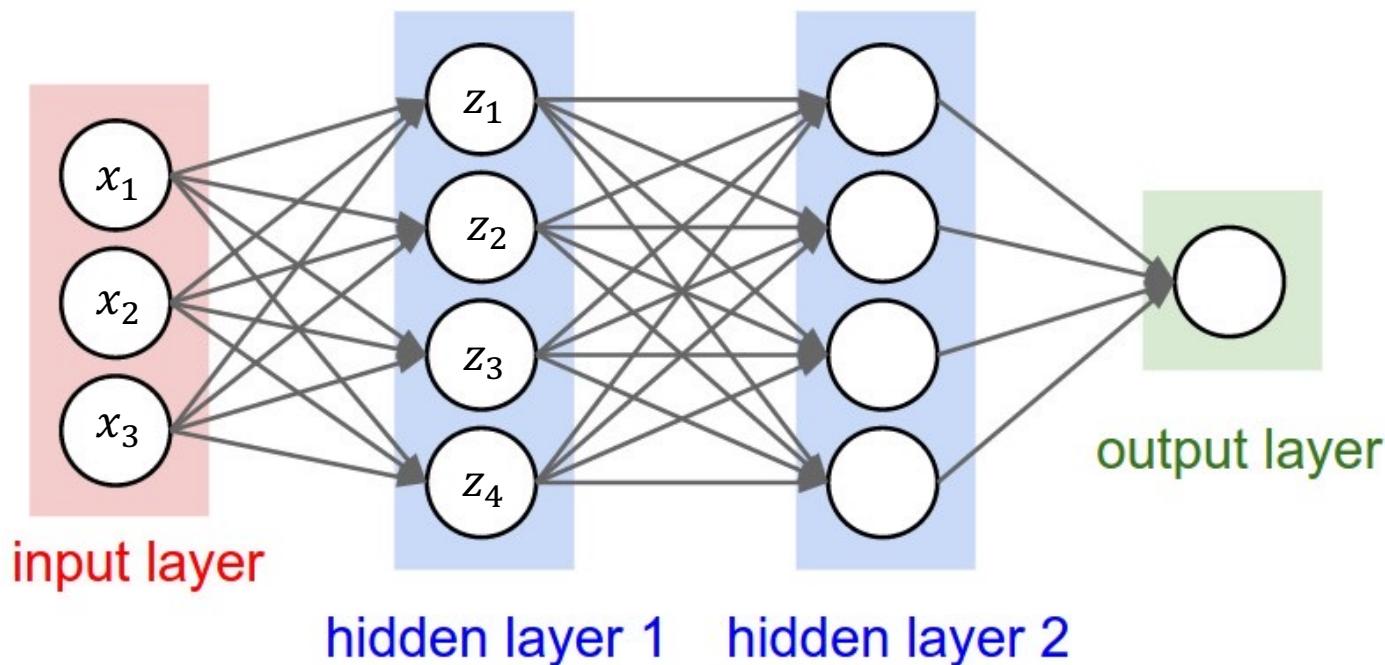
Полносвязный слой (fully connected, FC)

- На входе n чисел, на выходе m чисел
- x_1, \dots, x_n — входы
- z_1, \dots, z_m — выходы
- Каждый выход — линейная модель над входами

$$z_j = \sum_{i=1}^n w_{ji}x_i + b_j$$

Полносвязный слой (fully connected, FC)

$$z_j = \sum_{i=1}^n w_{ij} x_i$$



Полносвязный слой (fully connected, FC)

$$z_j = \sum_{i=1}^n w_{ji}x_i + b_j$$

- t линейных моделей, в каждой $(n + 1)$ параметров
- Всего примерно tn параметров в полносвязном слое

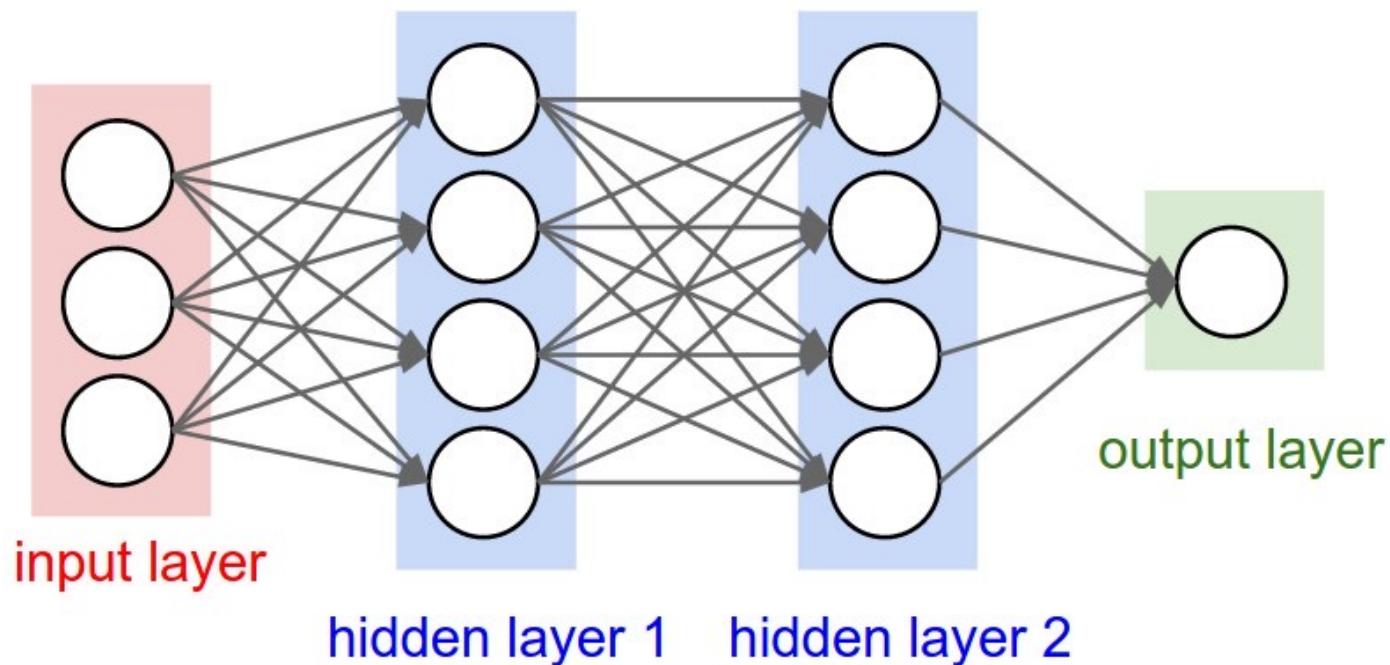
Полносвязный слой (fully connected, FC)

$$z_j = \sum_{i=1}^n w_{ji}x_i + b_j$$

- t линейных моделей, в каждой $(n + 1)$ параметров
- Всего примерно tn параметров в полносвязном слое
- Это очень много: если у нас 1.000.000 входных признаков и 1000 выходов, то это 1.000.000.000 параметров
- Надо много данных для обучения

Нелинейность

- Рассмотрим два полно связанных слоя



Нелинейность

- Рассмотрим два полносвязных слоя

$$\begin{aligned}s_k &= \sum_{j=1}^m v_{kj} z_j + c_k = \sum_{j=1}^m v_{kj} \sum_{i=1}^n w_{ji} x_i + \sum_{j=1}^m v_{kj} b_j + c_k = \\&= \sum_{j=1}^m \left(\sum_{i=1}^n v_{kj} w_{ji} x_i + v_{kj} b_j + \frac{1}{n} c_k \right)\end{aligned}$$

- То есть это ничем не лучше одного полносвязного слоя

Нелинейность

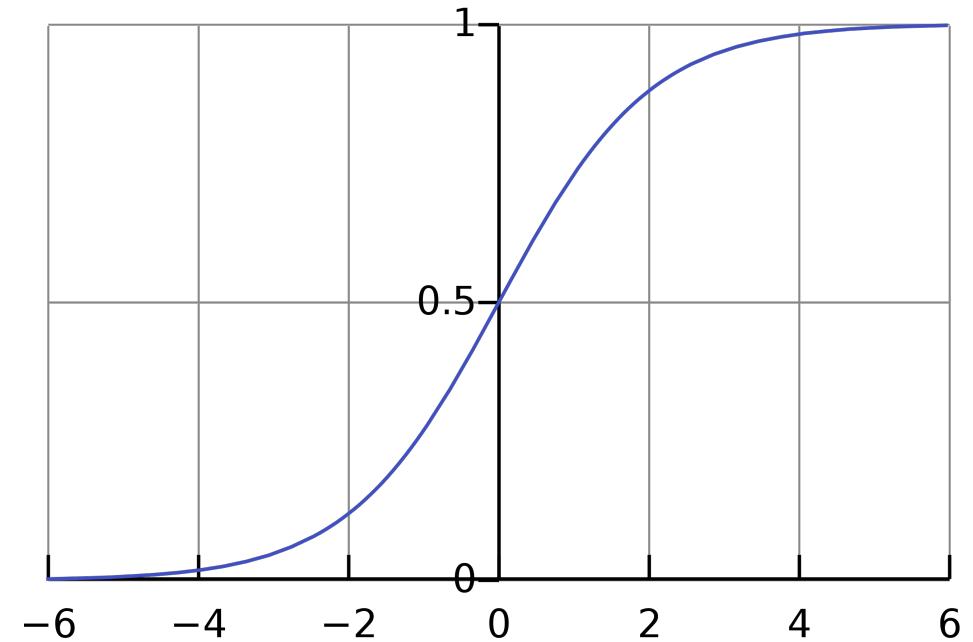
- Нужно добавлять нелинейную функцию после полносвязного слоя

$$z_j = f \left(\sum_{i=1}^n w_{ji} x_i + b_j \right)$$

Нелинейность

$$z_j = f \left(\sum_{i=1}^n w_{ji} x_i + b_j \right)$$

Вариант 1: $f(x) = \frac{1}{1+\exp(-x)}$
(сигмоида)

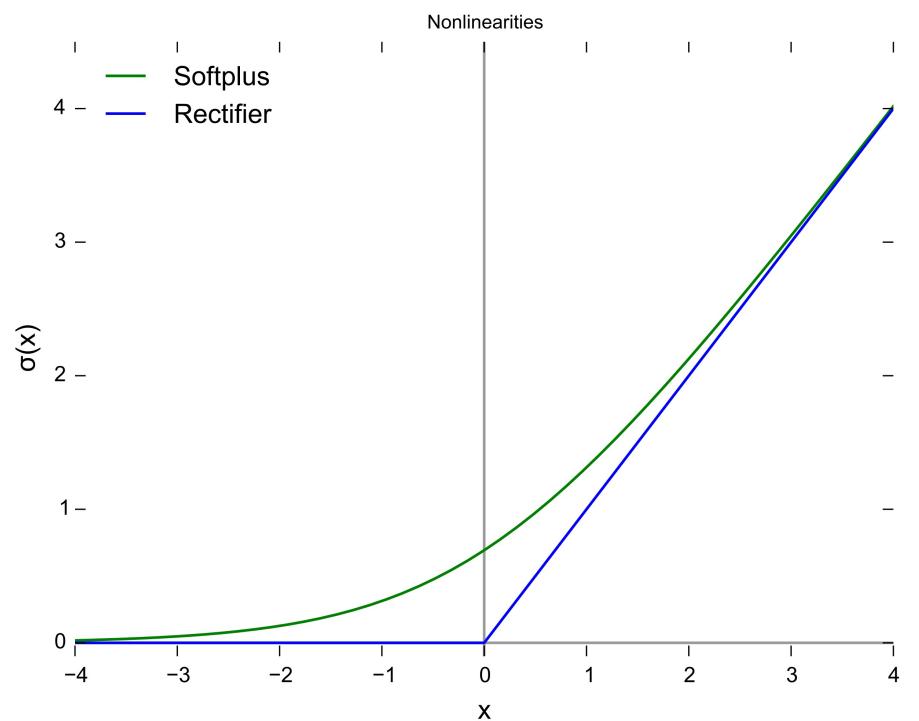


Нелинейность

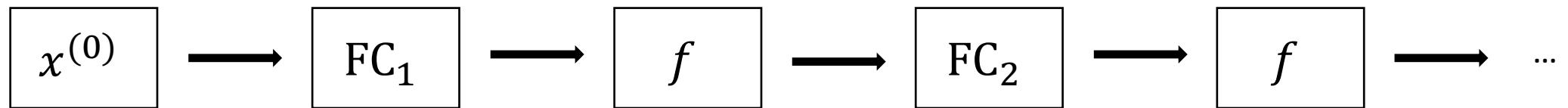
$$z_j = f \left(\sum_{i=1}^n w_{ji} x_i + b_j \right)$$

Вариант 2: $f(x) = \max(0, x)$

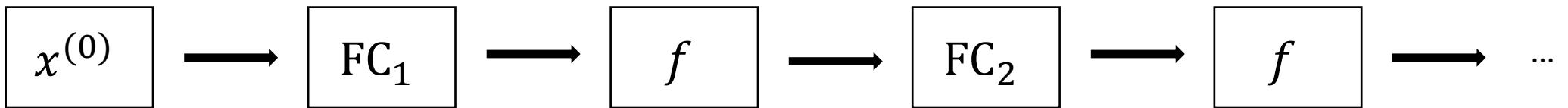
(ReLU, REctified Linear Unit)



Типичная полносвязная сеть



Типичная полносвязная сеть



- На входе признаки
- В последнем слое выходов столько, сколько целевых переменных мы предсказываем

Теорема Цыbenко

Вольное изложение:

- Пусть $g(x)$ — непрерывная функция
- Тогда можно построить двуслойную нейронную сеть, приближающую $g(x)$ с любой заранее заданной точностью

То есть двуслойные нейронные сети ОЧЕНЬ мощные!

Теорема Цыbenко

Вольное изложение:

- Пусть $g(x)$ — непрерывная функция
- Тогда можно построить двуслойную нейронную сеть, приближающую $g(x)$ с любой заранее заданной точностью

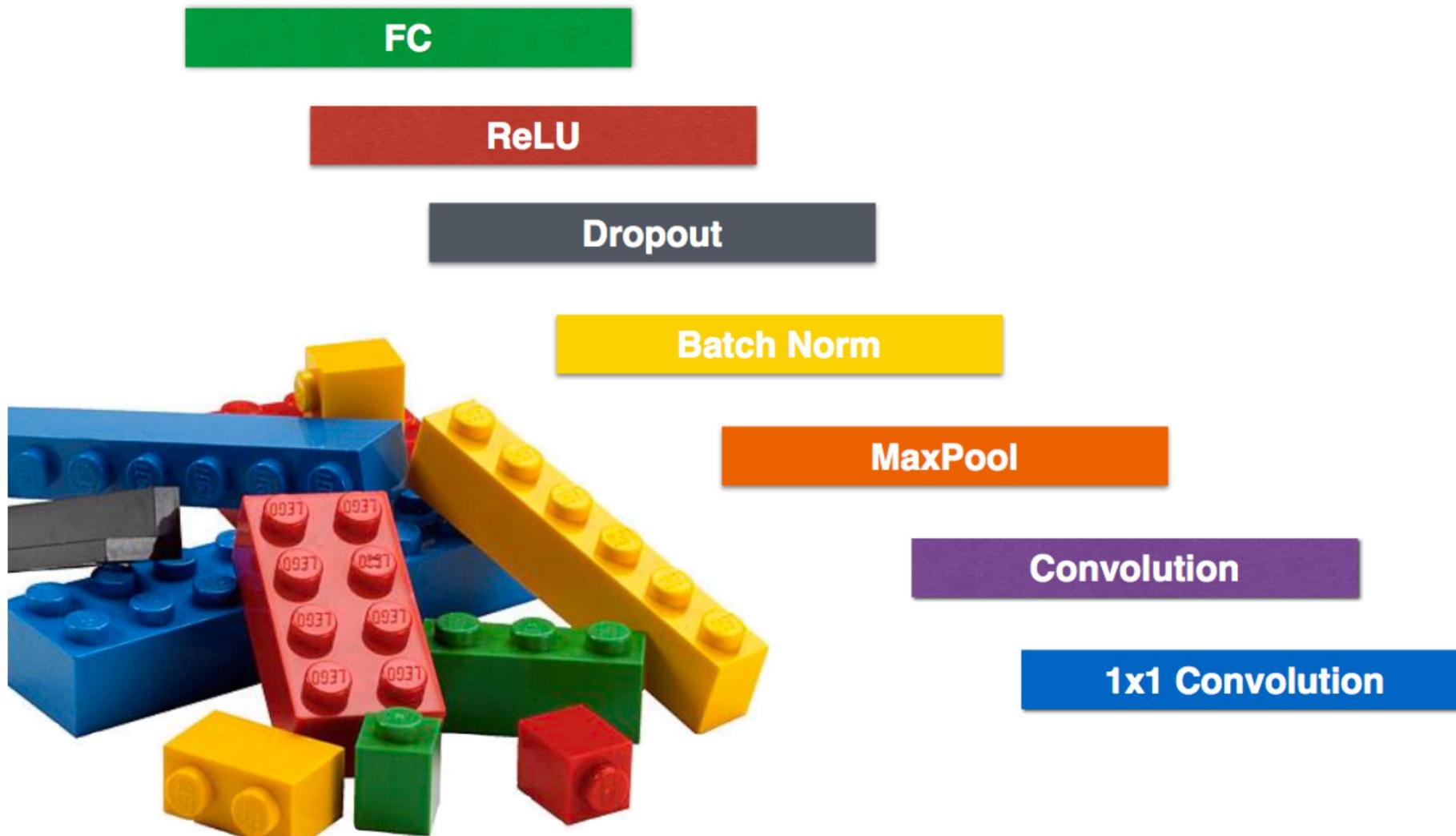
То есть двуслойные нейронные сети ОЧЕНЬ мощные!

Но очень много параметров и очень сложно обучать

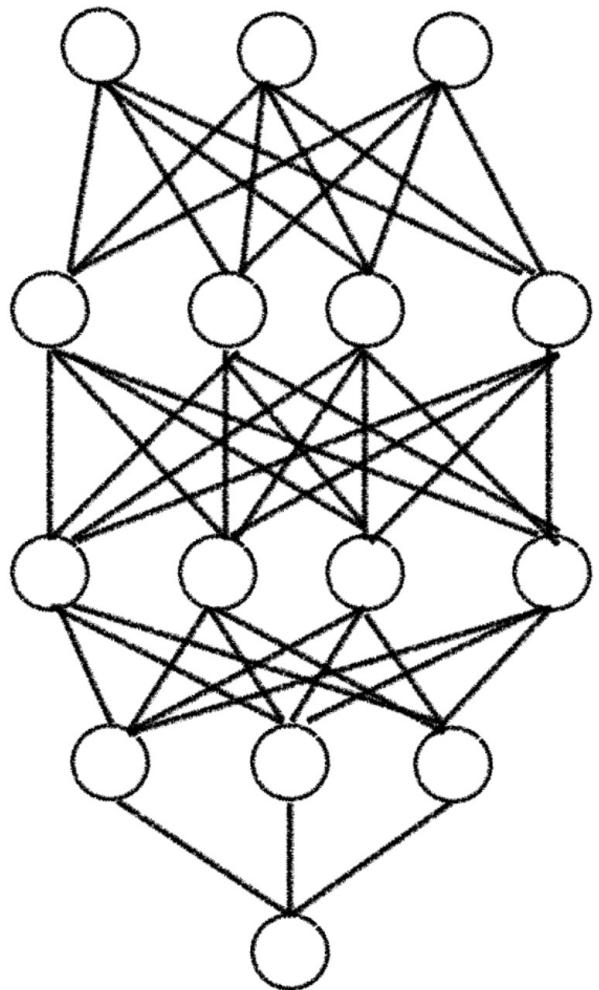
ФУНКЦИИ АКТИВАЦИИ

Название функции	Формула $f(x)$	Производная $f'(x)$
Логистический сигмоид σ	$\frac{1}{1+e^{-x}}$	$f(x)(1-f(x))$
Гиперболический тангенс \tanh	$\frac{e^x - e^{-x}}{e^x + e^{-x}}$	$1 - f^2(x)$
SoftSign	$\frac{x}{1+ x }$	$\frac{1}{(1+ x)^2}$
Ступенька (функция Хевисайда)	$\begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$	0
SoftPlus	$\log(1 + e^x)$	$\frac{1}{1+e^{-x}}$
ReLU	$\begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$	$\begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$
Leaky ReLU, Parameterized ReLU	$\begin{cases} ax, & x < 0 \\ x, & x \geq 0 \end{cases}$	$\begin{cases} a, & x < 0 \\ 1, & x \geq 0 \end{cases}$
ELU	$\begin{cases} \alpha(e^x - 1), & x < 0 \\ x, & x \geq 0 \end{cases}$	$\begin{cases} f(x) + \alpha, & x < 0 \\ 1, & x \geq 0 \end{cases}$

Нейросети – конструктор лего

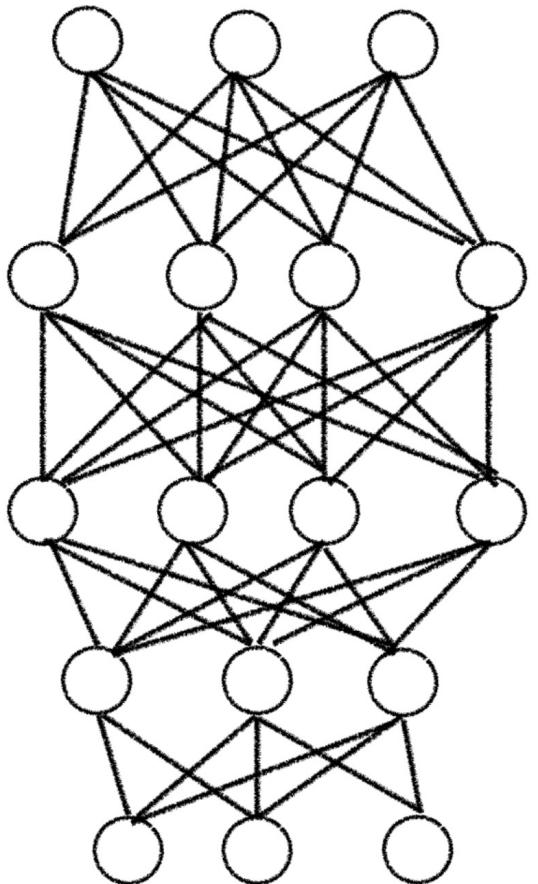


Регрессия



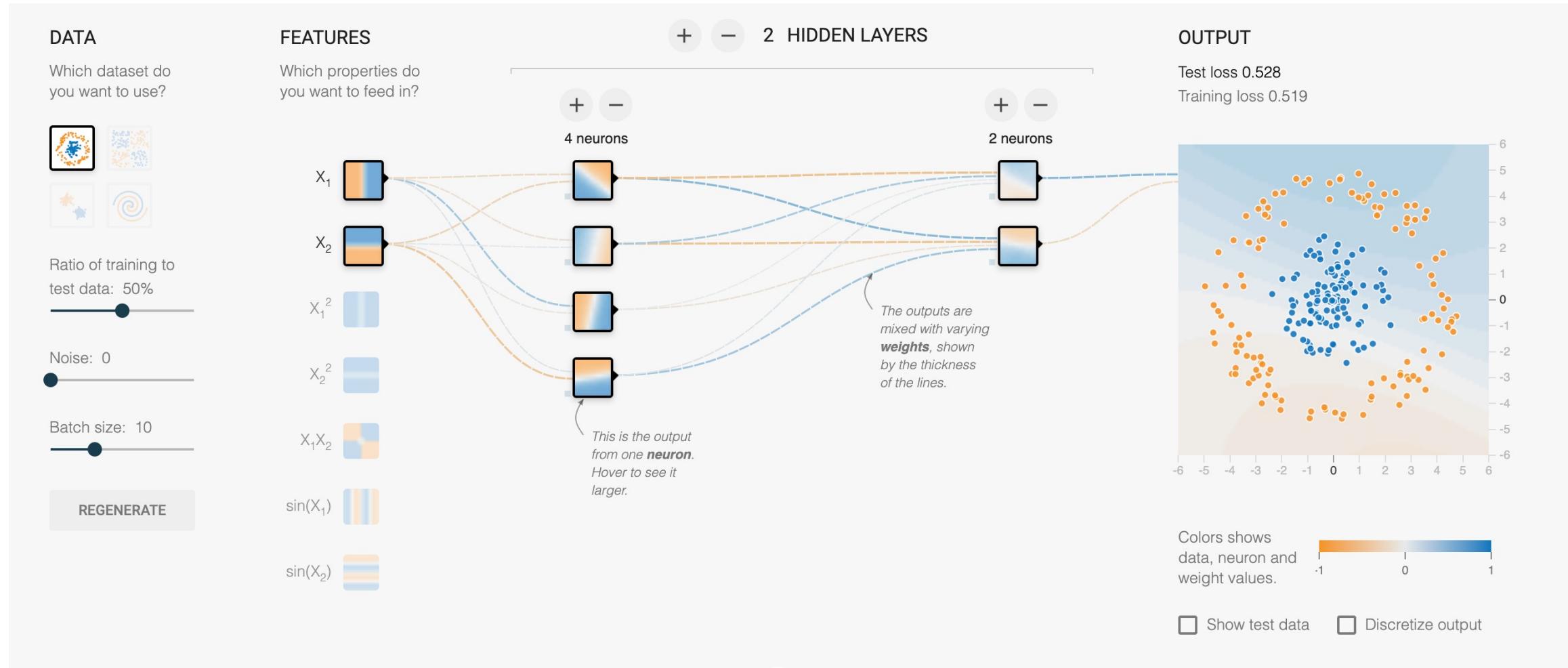
Input	
Fully connected layer (FC)	$XW + b$
ReLU	$\max(0, x)$
Dropout	$Bern(p)$
FC	$XW + b$
ReLU	$\max(0, x)$
FC	$XW + b$
Output	

Классификация



Input	
Fully connected layer (FC)	$XW + b$
ReLU	$\max(0, x)$
Dropout	$Bern(p)$
FC	$XW + b$
ReLU	$\max(0, x)$
FC	$XW + b$
Softmax	$\text{softmax}(x)$
Output	

Пограться



Фреймворки

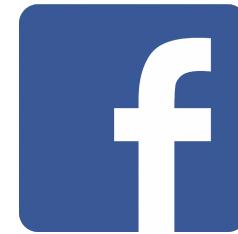
theano

TensorFlow

P Y T  R C H



Google



Обертки



<https://www.youtube.com/watch?v=ghZyptkanB0>

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:

$$w^t = w^{t-1} - \eta \nabla Q(w^{t-1})$$

Новая точка

Размер шага

Градиент в
предыдущей
точке

Сходимость

- Останавливаем процесс, если

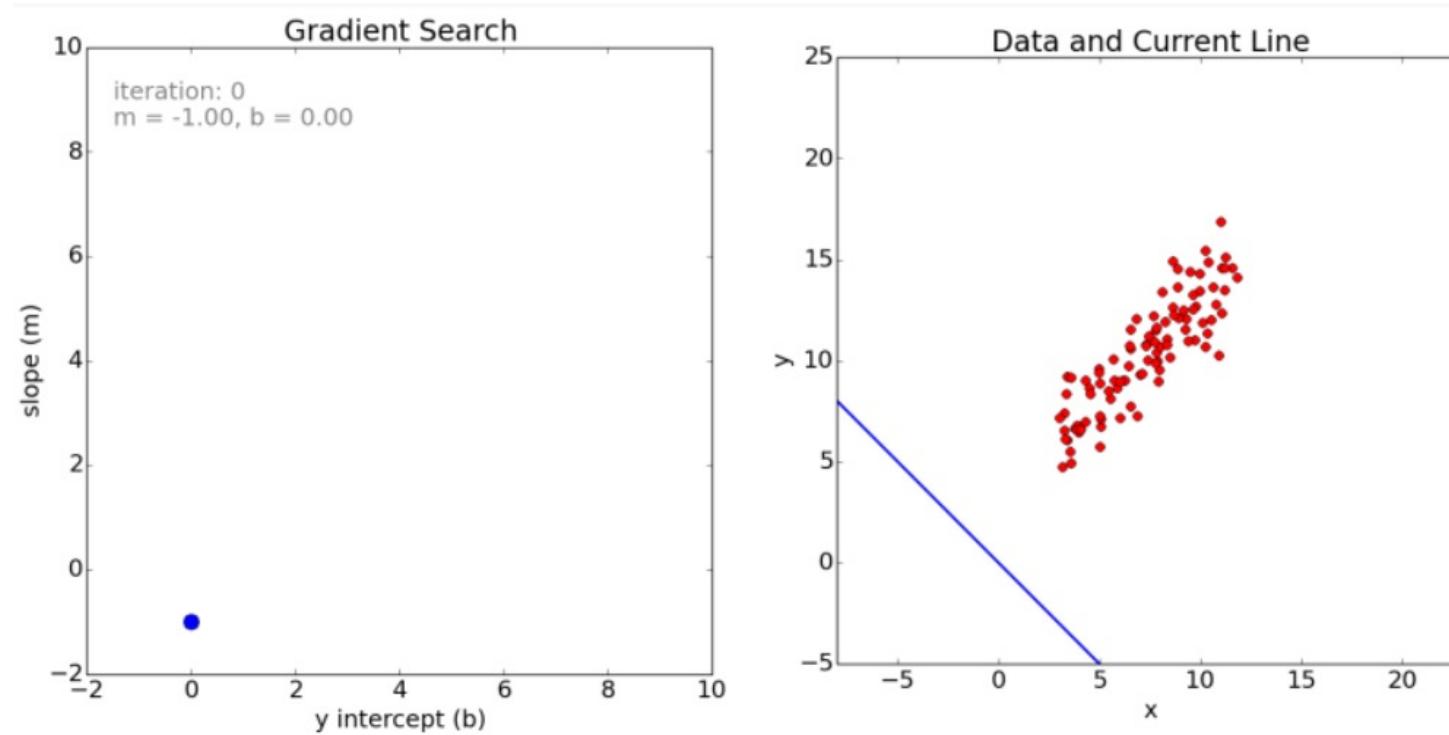
$$\|w^t - w^{t-1}\| < \varepsilon$$

- Другой вариант:

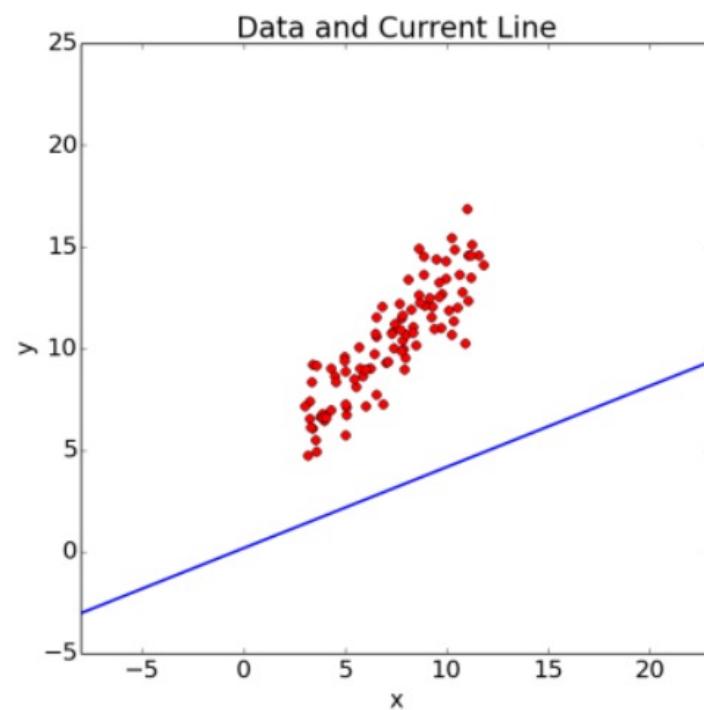
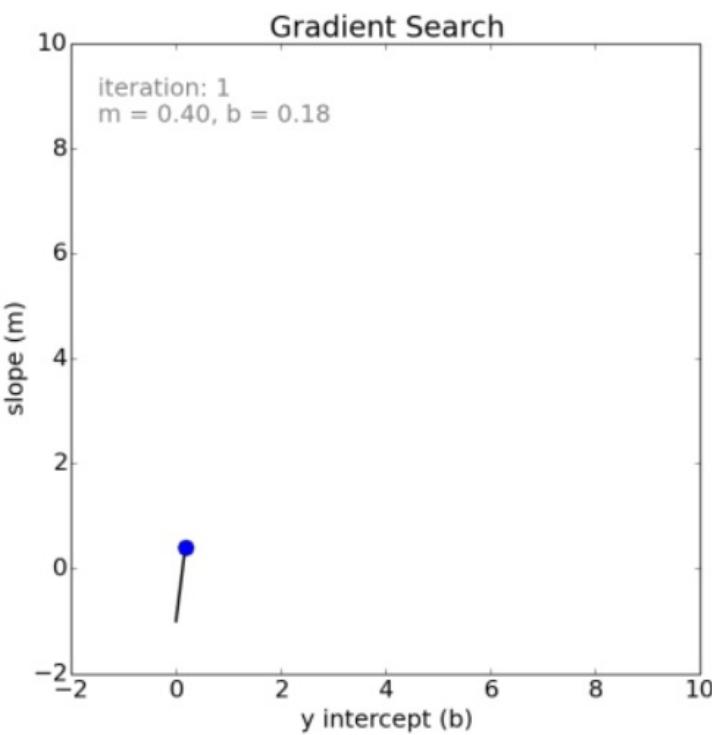
$$\|\nabla Q(w^t)\| < \varepsilon$$

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

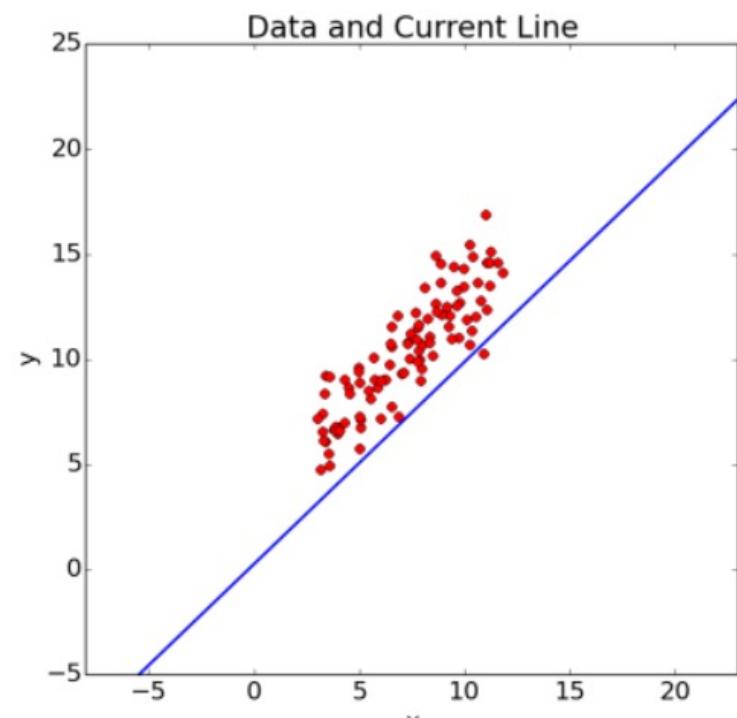
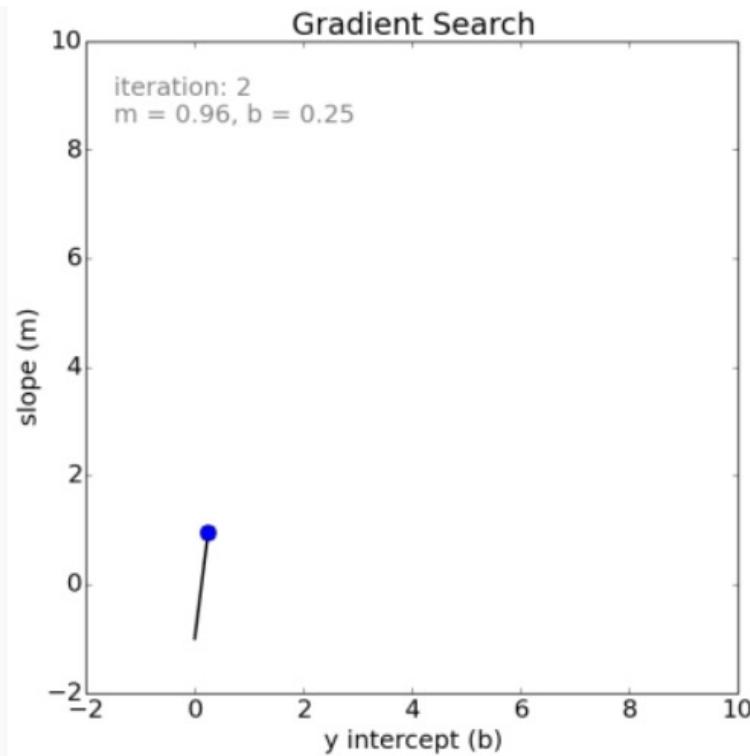
Парная регрессия



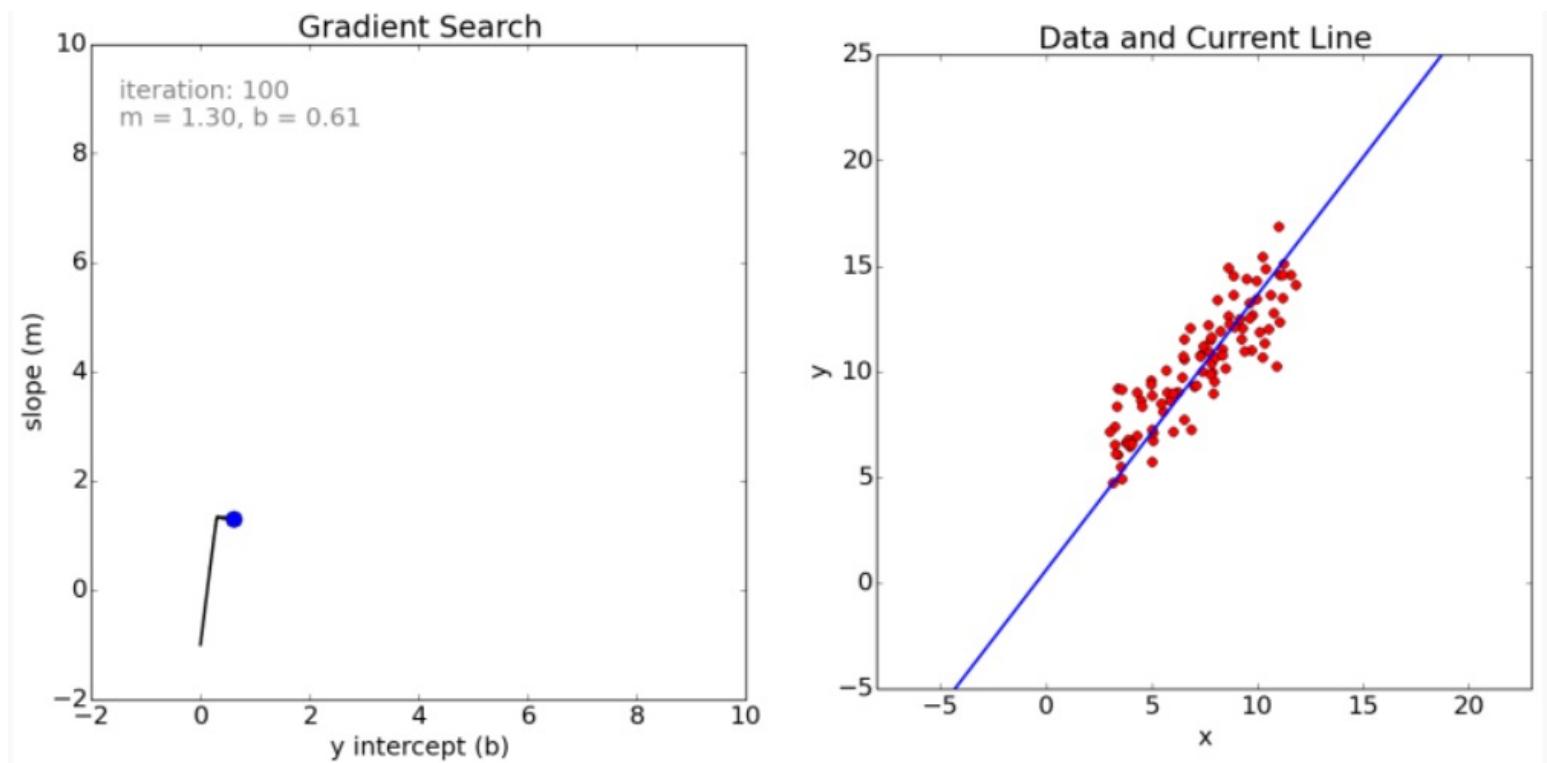
Парная регрессия



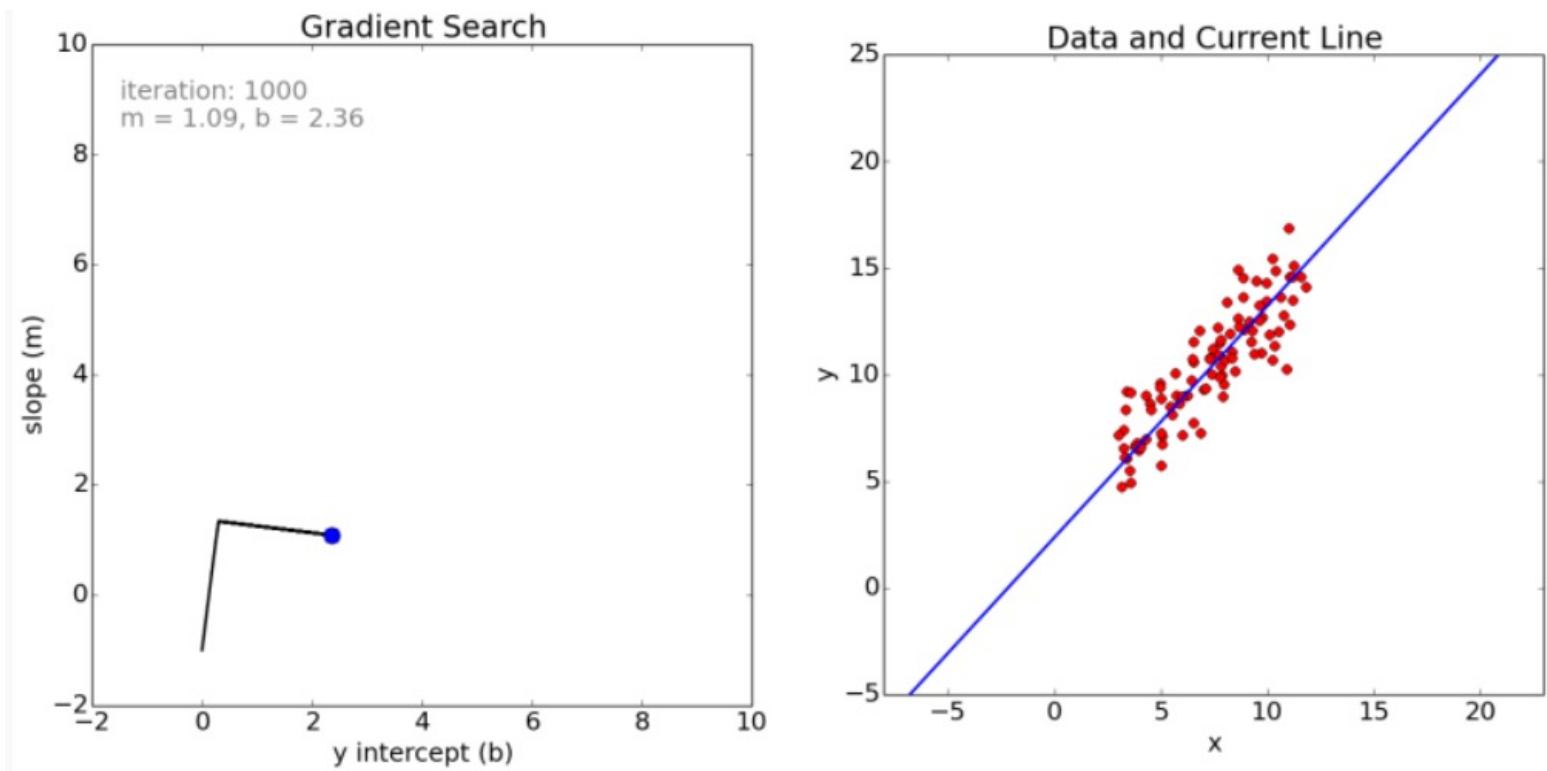
Парная регрессия



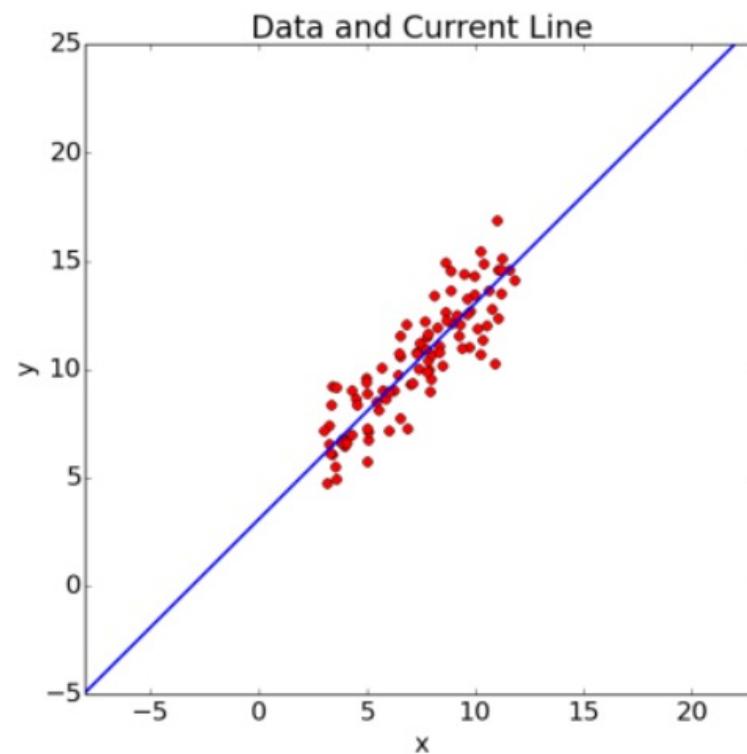
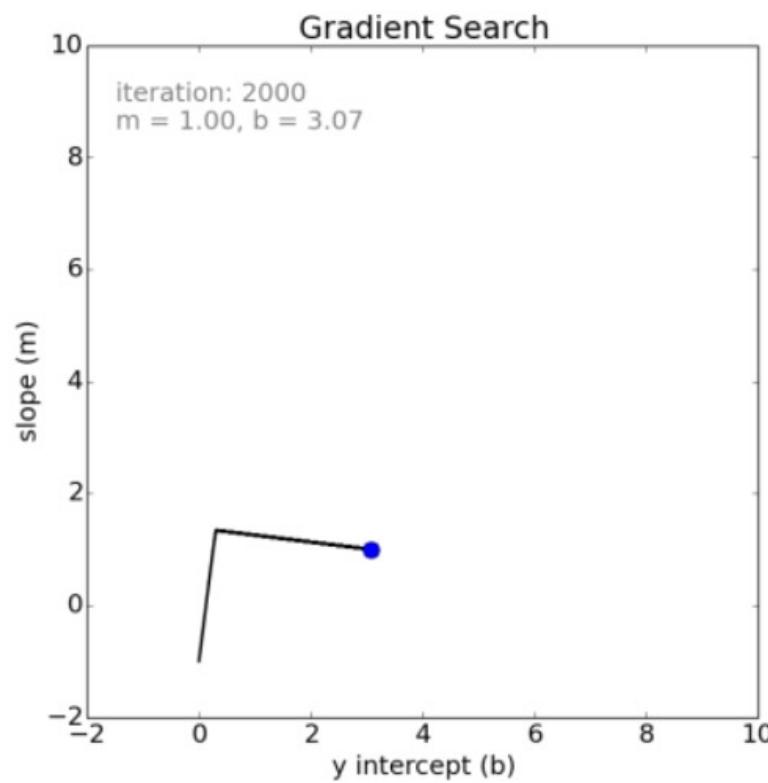
Парная регрессия



Парная регрессия



Парная регрессия



Градиентный спуск

1. Начальное приближение: w^0

2. Повторять:

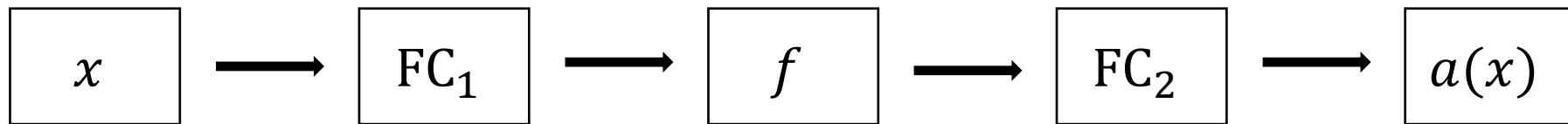
$$w^t = w^{t-1} - \eta \nabla Q(w^{t-1})$$

3. Останавливаемся, если

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

Обучение нейронных сетей

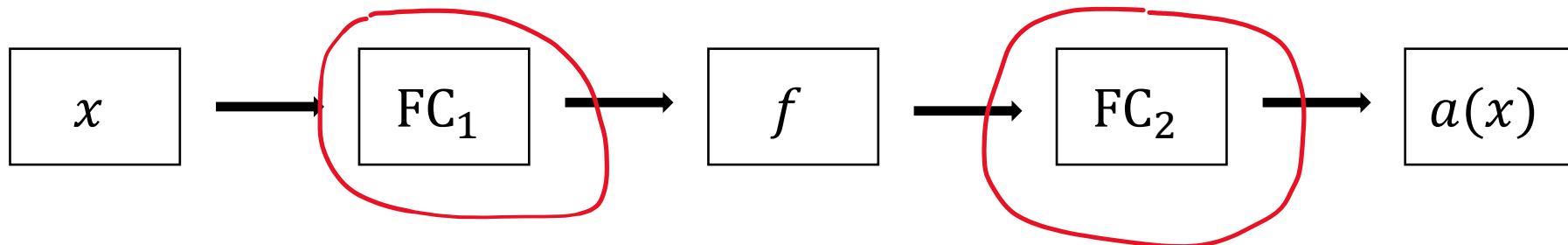
- Все слои обычно дифференцируемы, поэтому можно посчитать производные по всем параметрам



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

Обучение нейронных сетей

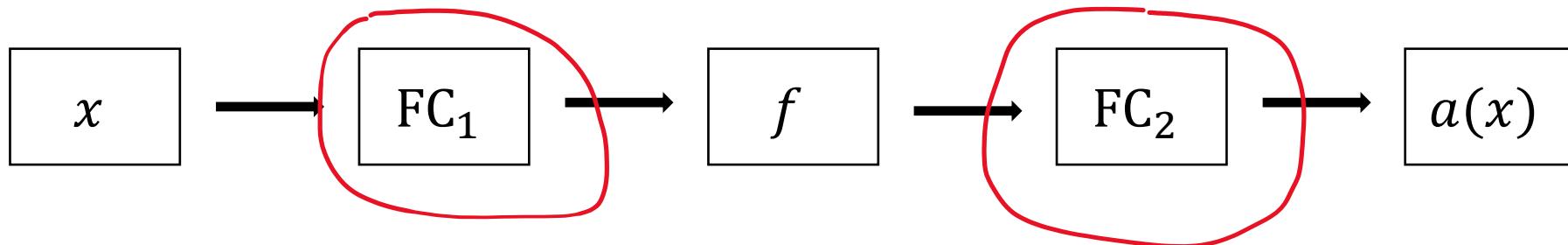
- Все слои обычно дифференцируемы, поэтому можно посчитать производные по всем параметрам



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

Обучение нейронных сетей

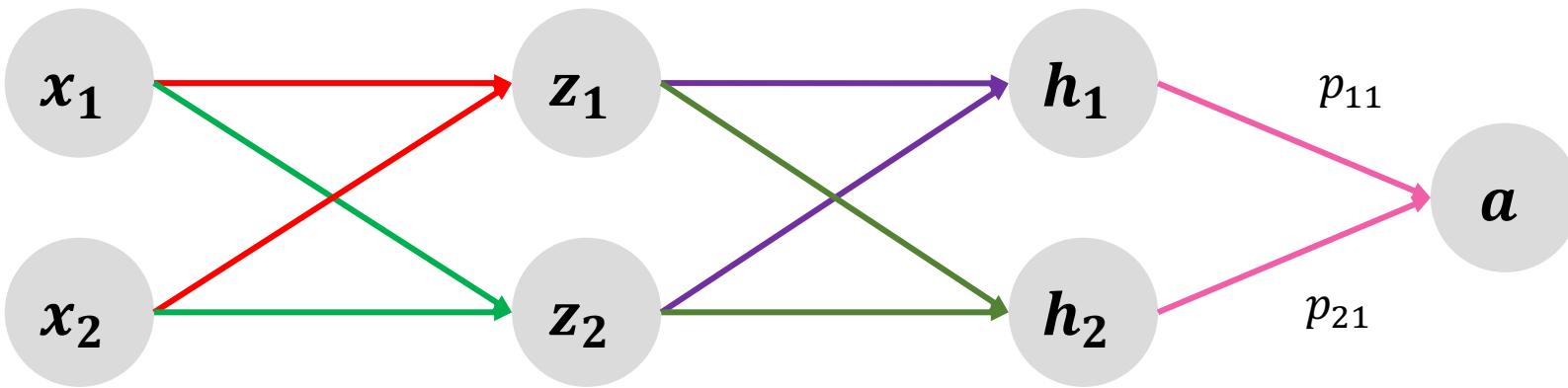
- Все слои обычно дифференцируемы, поэтому можно посчитать производные по всем параметрам



- $a(x) = FC_2(f(FC_1(x)))$

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

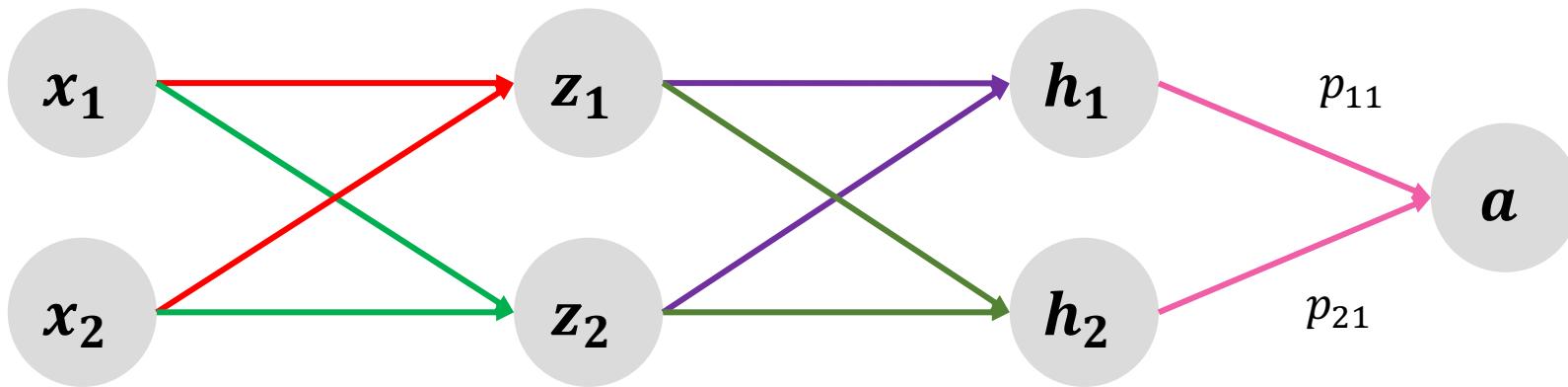
Как считать производные?



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

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

Как считать производные?

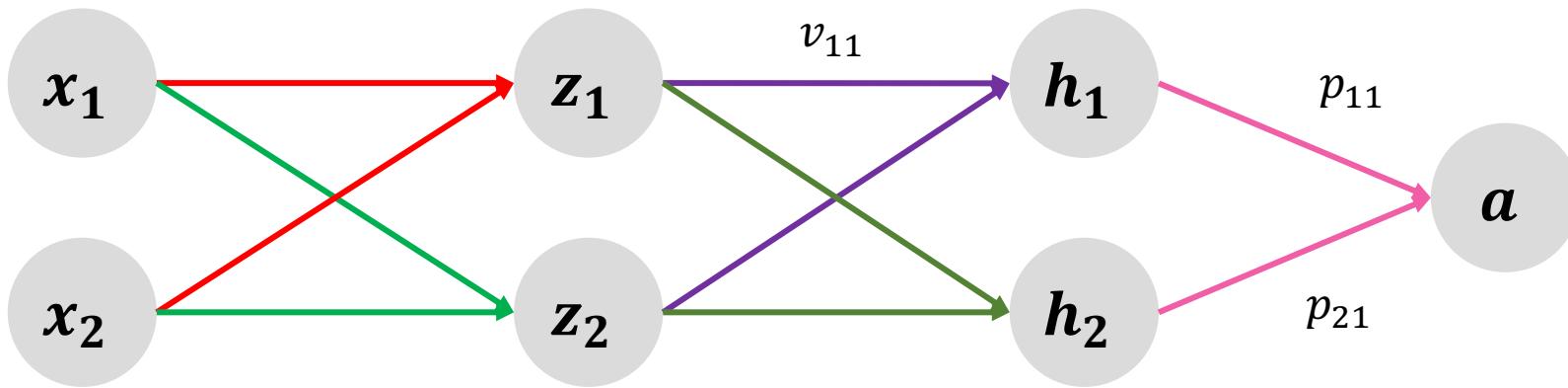


$$a(x) = p_{11}h_1(x) + p_{21}h_2(x)$$

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

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

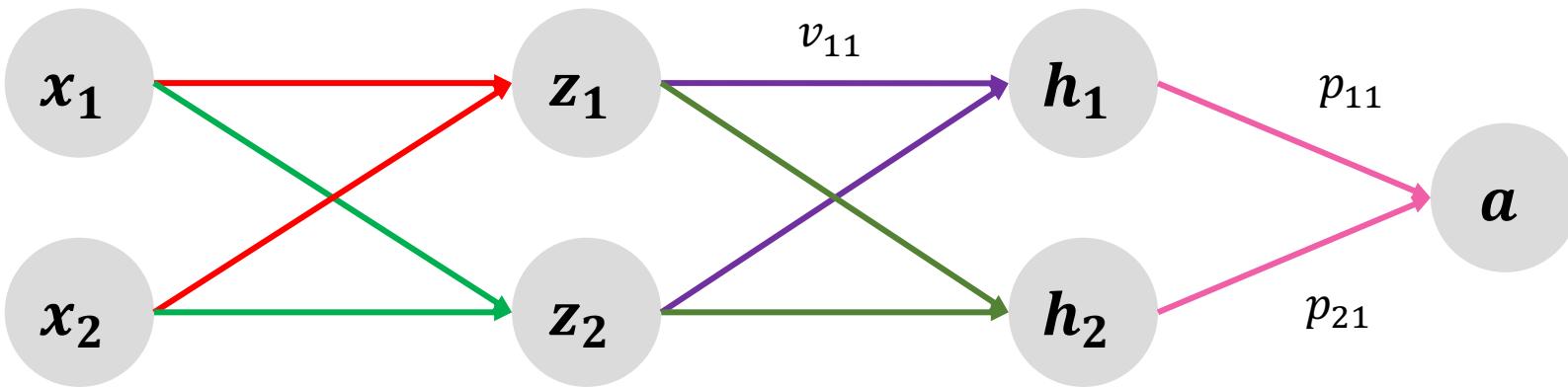
Как считать производные?



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

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

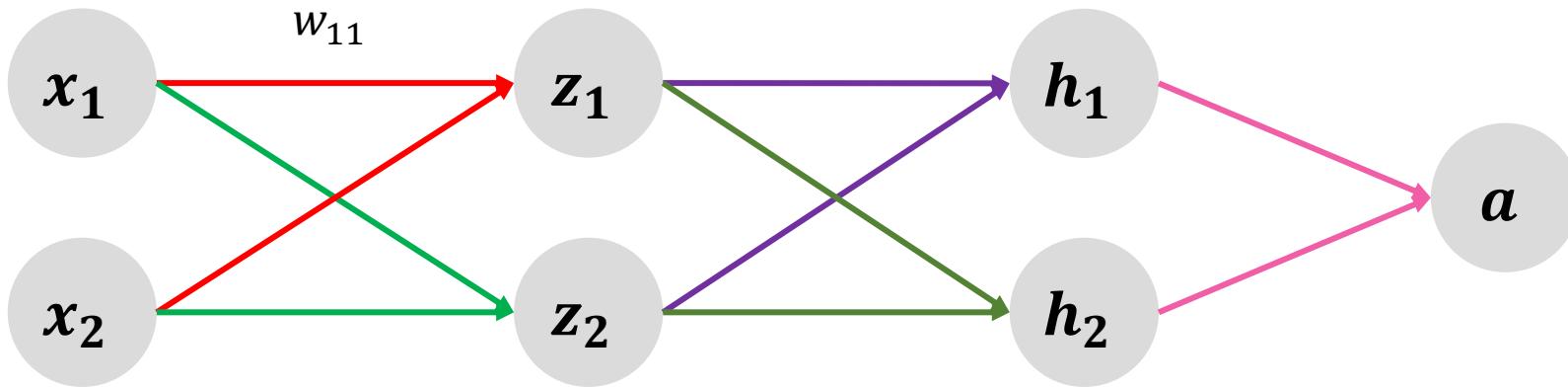
Как считать производные?



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

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

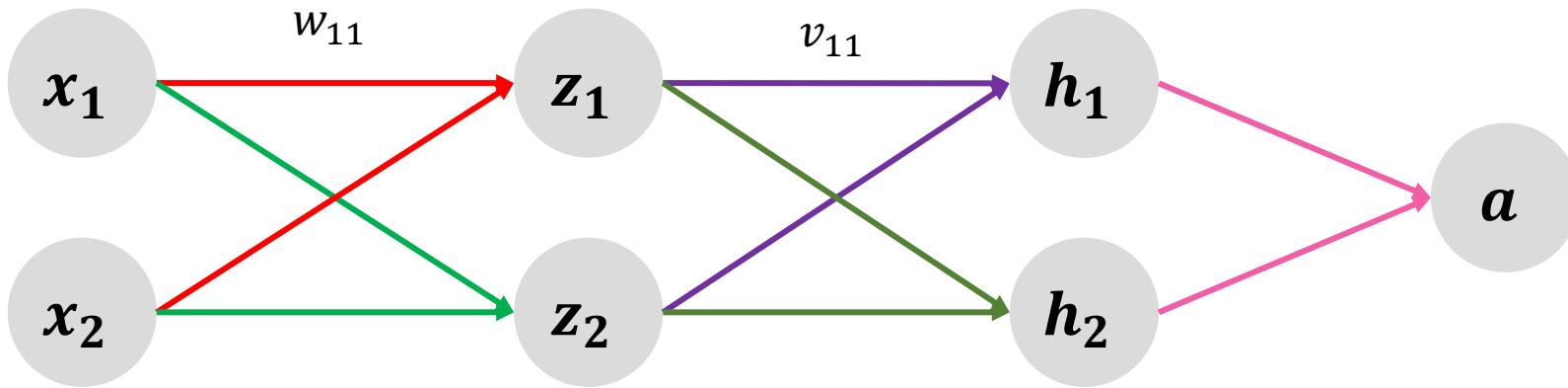
Как считать производные?



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

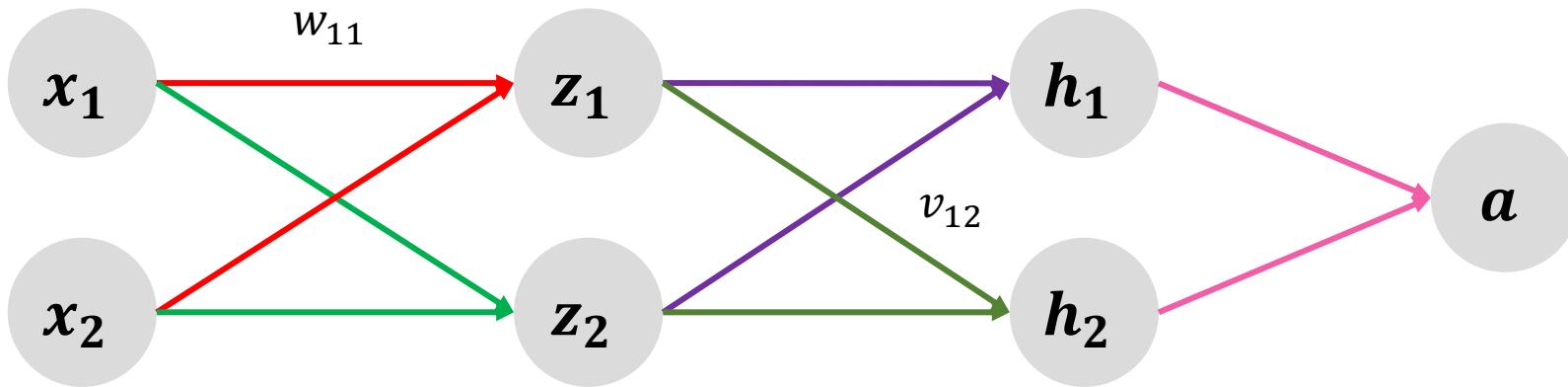
- Показывает, как сильно изменится a при изменении w_{11}

Как считать производные?



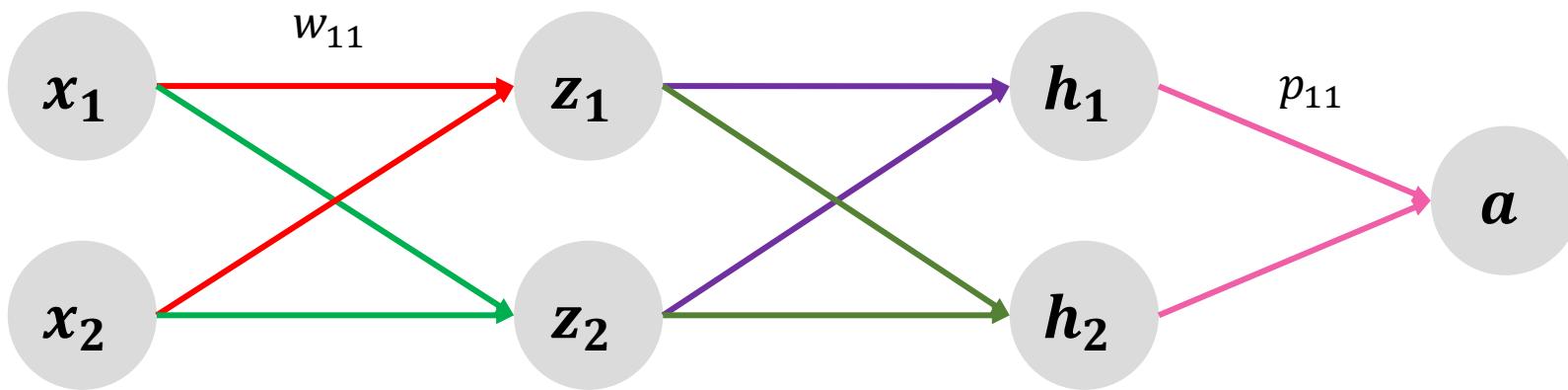
- Как сильно изменится a при изменении w_{11} ?
- Влияет ли на это v_{11} ?

Как считать производные?



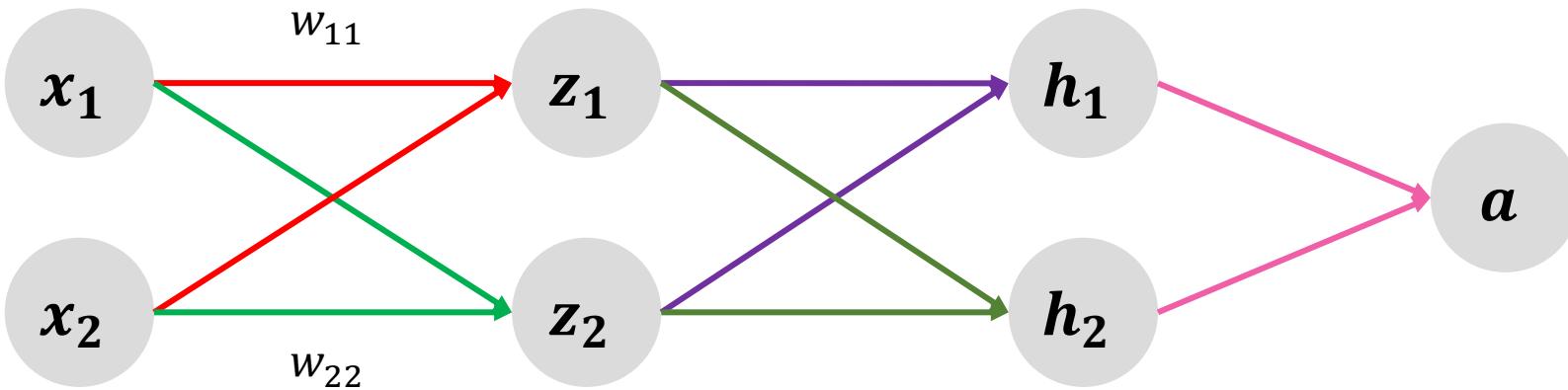
- Как сильно изменится a при изменении w_{11} ?
- Влияет ли на это v_{12} ?

Как считать производные?



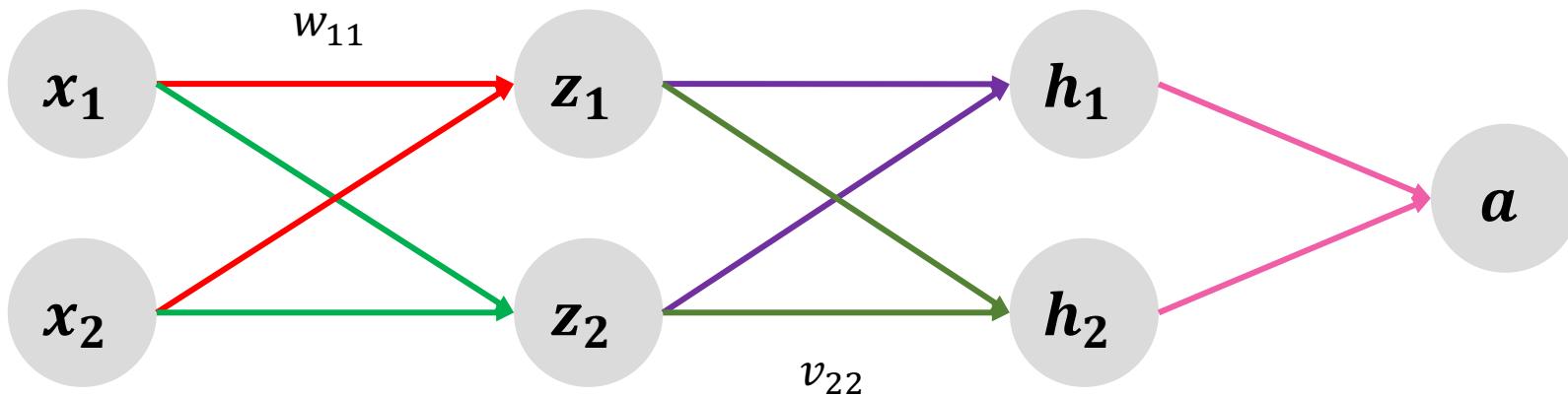
- Как сильно изменится a при изменении w_{11} ?
- Влияет ли на это p_{11} ?

Как считать производные?



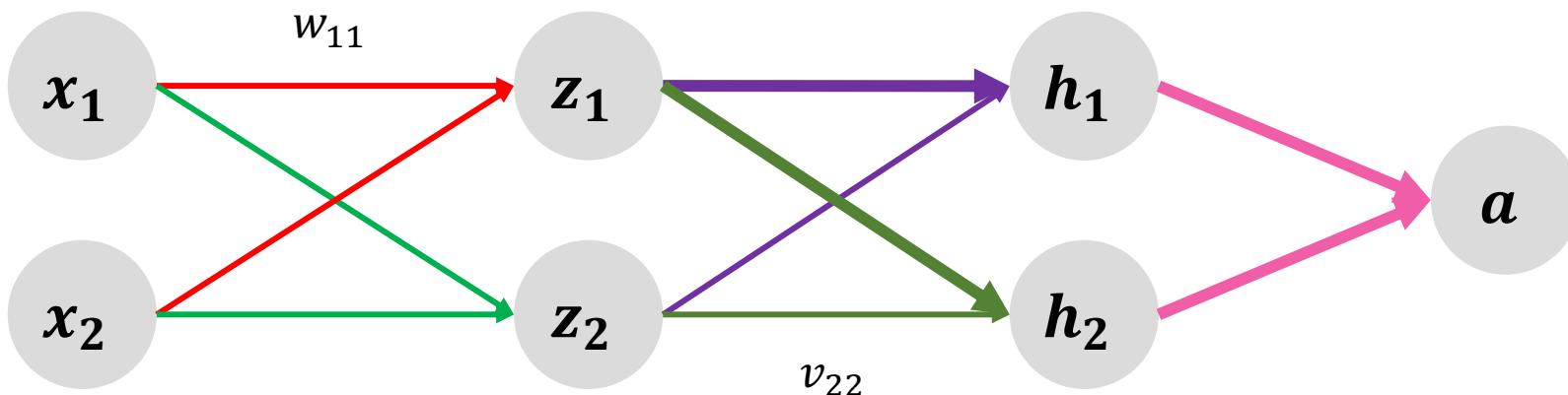
- Как сильно изменится a при изменении w_{11} ?
- Влияет ли на это w_{22} ?

Как считать производные?



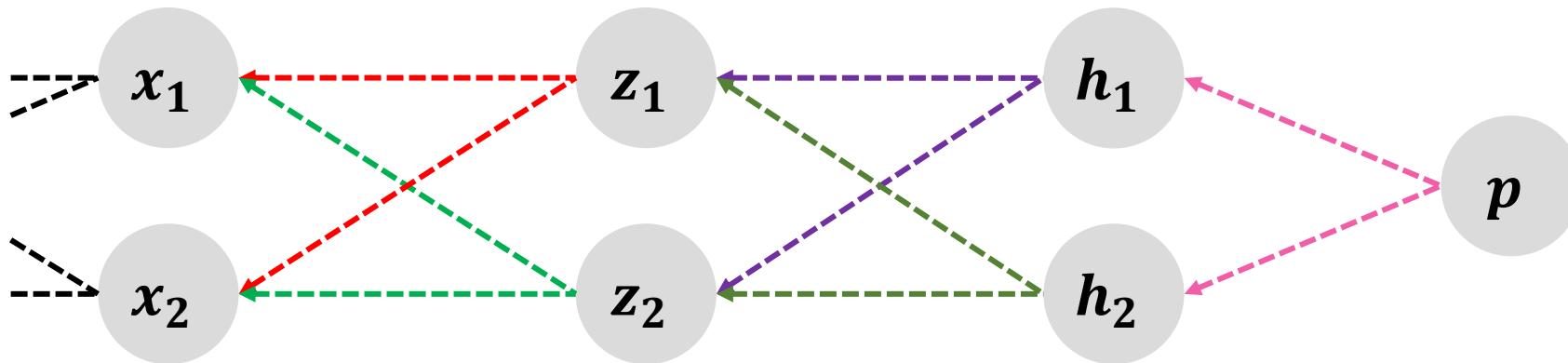
- Как сильно изменится a при изменении w_{11} ?
- Влияет ли на это v_{22} ?

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?



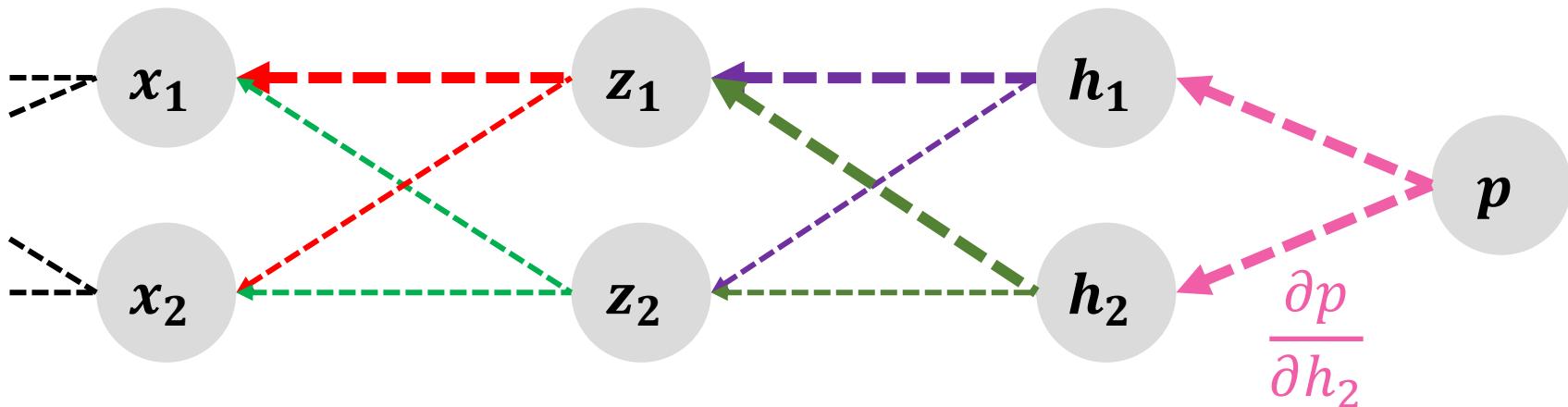
- Мы как бы идём в обратную сторону по графу и считаем производные
- Метод обратного распространения ошибки (backpropagation)

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

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

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

$$1: \frac{\partial p}{\partial x_2} = \frac{\partial p}{\partial h_1} \frac{\partial h_1}{\partial z_1} \frac{\partial z_1}{\partial x_2} + \frac{\partial p}{\partial h_2} \frac{\partial h_2}{\partial z_1} \frac{\partial z_1}{\partial x_2} + \frac{\partial p}{\partial h_1} \frac{\partial h_1}{\partial z_2} \frac{\partial z_2}{\partial x_2} + \frac{\partial p}{\partial h_2} \frac{\partial h_2}{\partial z_2} \frac{\partial z_2}{\partial x_2}$$



Backprop

- Во многие формулы входят одни и те же производные
- В backprop каждая частная производная вычисляется один раз