Введение в глубинное обучение

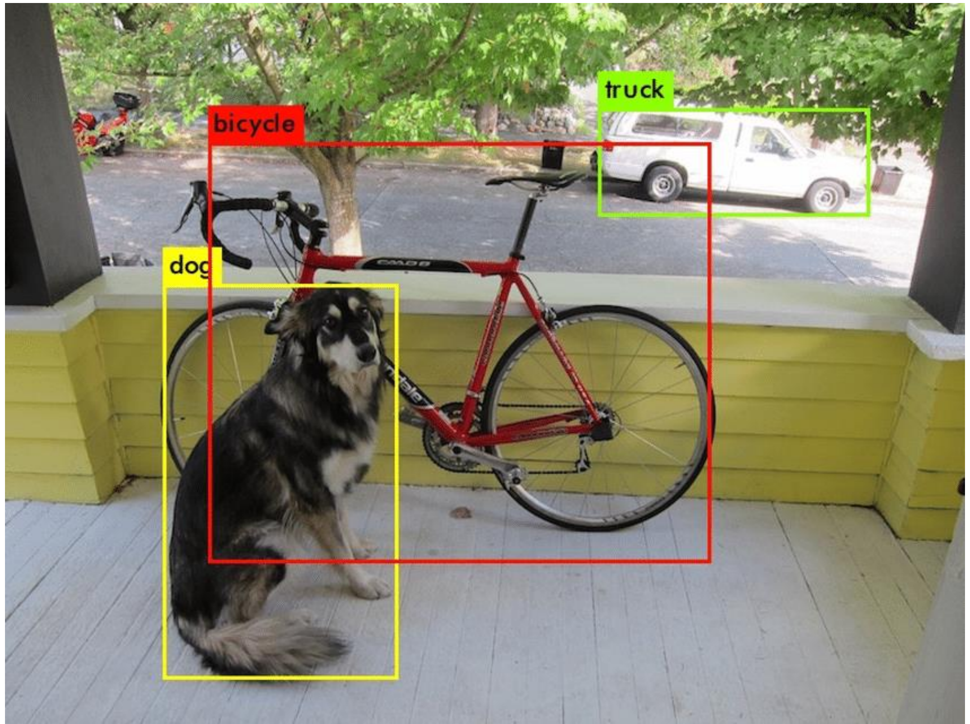
Евгений Соколов

esokolov@hse.ru

Сириус, декабрь 2024

Пример задачи: детекция объектов

- Дана фотография
- Найти человека на фотографии
- Гарантируется, что человек точно есть и точно один



Обозначения

- x — объект — что анализируем
  - Фотография
  - Массив $n * m * 3$
- y — ответ, целевая переменная — что на выходе
  - Прямоугольник, содержащий человека
  - Четыре числа: координаты левого верхнего угла прямоугольника (a, b), ширина и высота (c, d)
  - $y = (a, b, c, d)$

Обучающая выборка

- Нам нужны примеры, из которых будем выводить правила

- $X = (x_i, y_i)_{i=1}^{\ell}$ — обучающая выборка
- ℓ — размер выборки

- Много фотографий с прямоугольниками

Признаки

- Объекты — абстрактные сущности
- Компьютеры работают только с числами

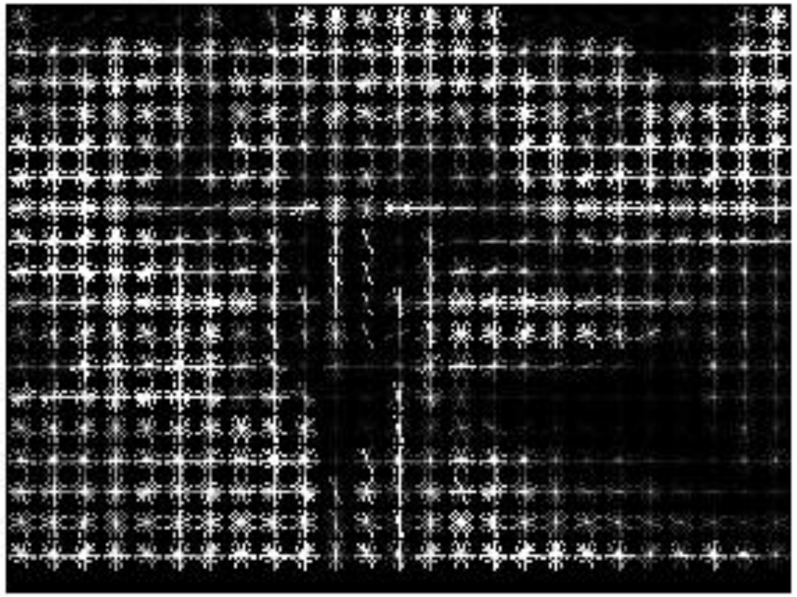
- Признаки, факторы — характеристики объектов
- d — количество признаков
- $x = (x_1, \ldots, x_d)$ — признаковое описание

Признаки

Признаки

Input image

Histogram of Oriented Gradients



Алгоритм

- $a(x)$ — алгоритм, модель — функция, оценивающая ответ для любого объекта
- Линейная модель: $a(x) = w_0 + w_1 x_1 + \dots + w_d x_d$

Функция потерь

- Как понять, что наш алгоритм работает?
- Сравнить его прогнозы с правильными ответами на обучающей выборке!

$$L(y,z) = L\left(\left(a_y, b_y, c_y, d_y\right), \left(a_z, b_z, c_z, d_z\right)\right) =$$
$$\left(a_y - a_z\right)^2 + \left(b_y - b_z\right)^2 + \left(c_y - c_z\right)^2 + \left(d_y - d_z\right)^2$$

Функционал ошибки

- Функционал ошибки — мера качества работы алгоритма на выборке
- Обычно вычисляется как среднее значение функции потерь:

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

- Чем меньше, тем лучше

Функционал ошибки

- Должен соответствовать бизнес-требованиям
- Одна из самых важных составляющих анализа данных

Обучение алгоритма

- Есть обучающая выборка и функционал ошибки
- Семейство алгоритмов $\mathcal{A}$
  - Из чего выбираем алгоритм
  - Пример: все линейные модели
  - $\mathcal{A} = \{w_0 + w_1 x_1 + \dots + w_d x_d \mid w_0, w_1, \dots, w_d \in \mathbb{R}\}$
- Обучение: поиск оптимального алгоритма с точки зрения функционала ошибки

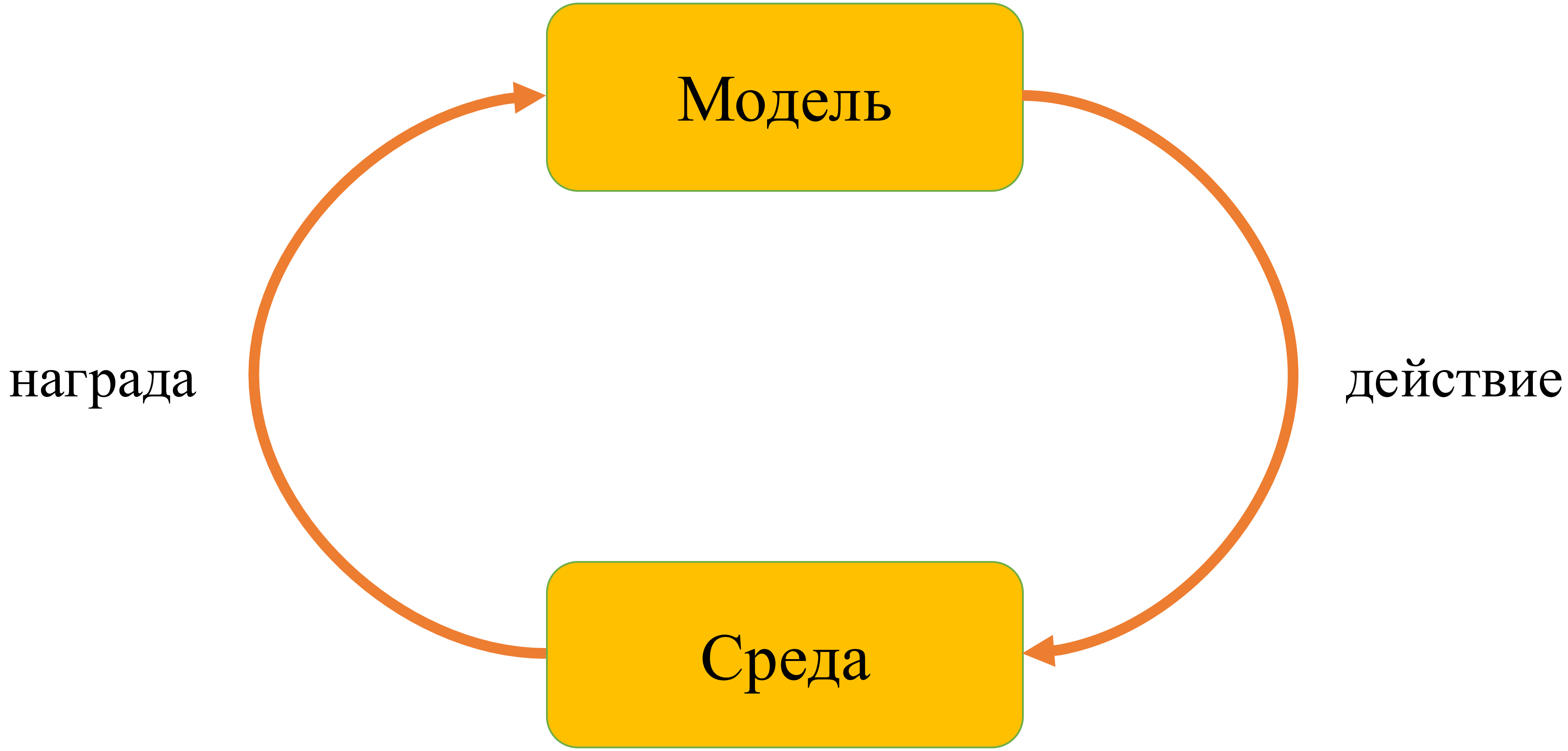
$$a(x) = \arg\min_{a \in \mathcal{A}} Q(a, X)$$

Машинное обучение

- Не все задачи имеют такую формулировку!

- Обучение без учителя
- Обучение с подкреплением
- И т.д.

Обучение с подкреплением



Резюме

- Машинное обучение — построение алгоритмов на основе примеров
- Постановка задачи: объект и ответ
- Обучающая выборка
- Модель/алгоритм
- Функция потерь

Обобщающая способность

Обобщающая способность

Как готовиться к экзамену?

Заучить все примеры с занятий	Разобраться в предмете и усвоить алгоритмы решения задач

Обобщающая способность

Как готовиться к экзамену?

Заучить все примеры с занятий	Разобраться в предмете и усвоить алгоритмы решения задач
Переобучение (overfitting)	Обобщение (generalization)

Обобщающая способность

Как готовиться к экзамену?

Заучить все примеры с занятий

Переобучение (overfitting)

Хорошее качество на обучении
Низкое качество на новых данных

Разобраться в предмете и усвоить алгоритмы решения задач

Обобщение (generalization)

Хорошее качество на обучении
Хорошее качество на новых данных

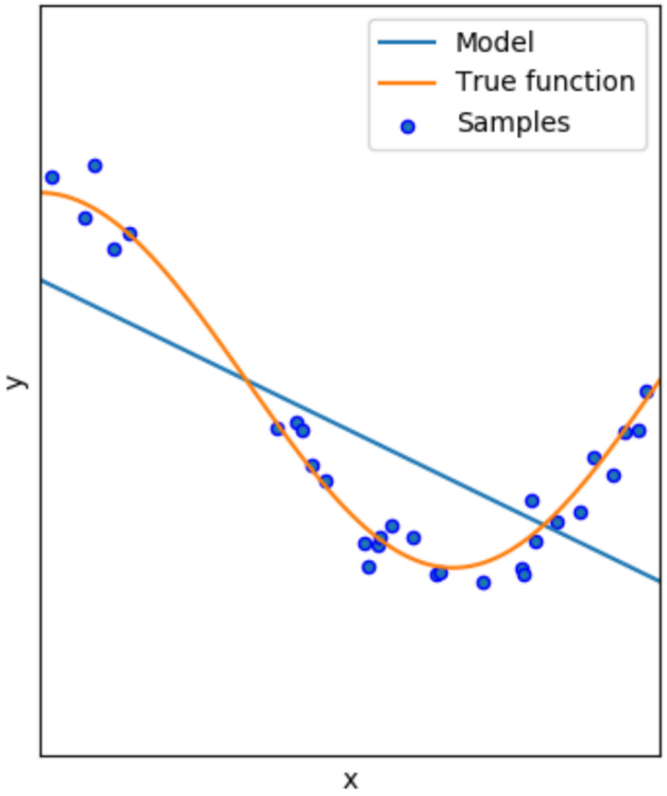
Отложенная выборка

Отложенная выборка

- Слишком большое обучение — тестовая выборка нерепрезентативна
- Слишком большой тест — модель не сможет обучиться
- Обычно: 70/30, 80/20

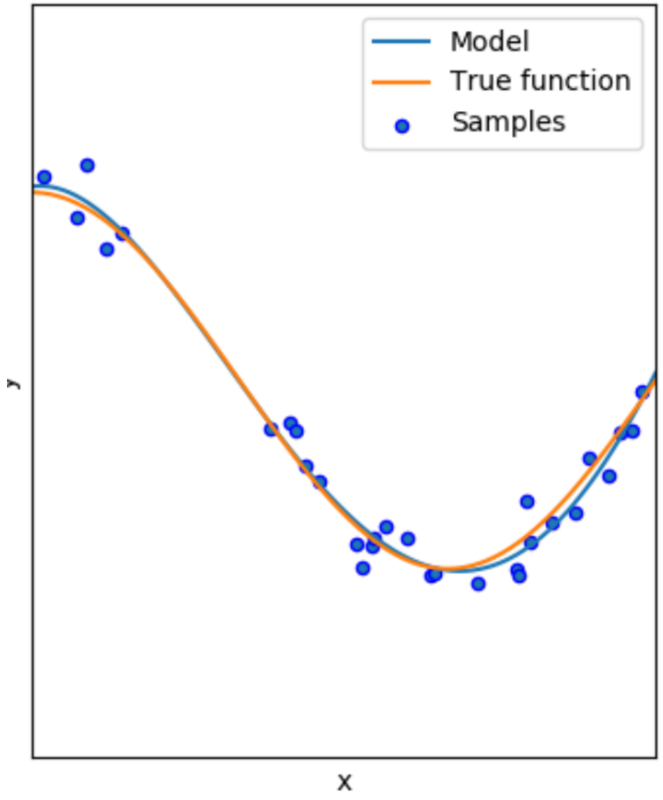
Нелинейная задача

$$a(x) = w_0 + w_1 x$$



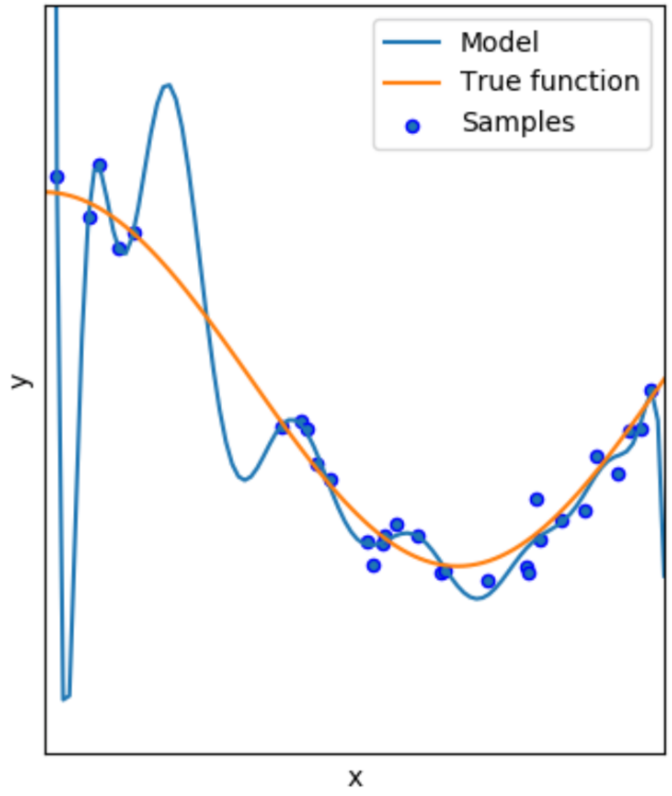
Нелинейная задача

$$a(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4$$



Нелинейная задача

$$a(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4 + \cdots + w_{15} x^{15}$$



Симптом переобучения

$$a(x) = 0.5 + 13458922x - 43983740x^2 + \cdots$$

- Большие коэффициенты — симптом переобучения
- Эмпирическое наблюдение

Градиентные методы

Градиент

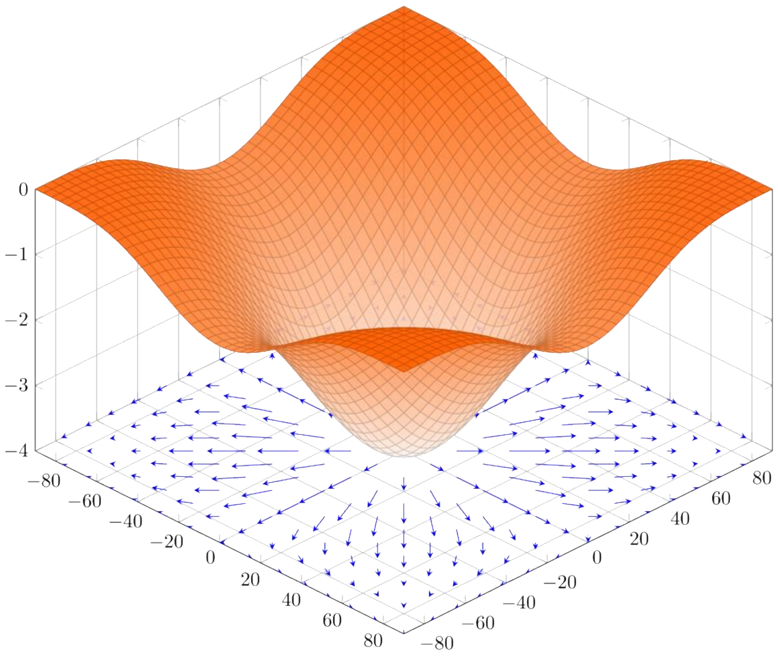
- Градиент — вектор частных производных

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_d}\right)$$

- У градиента есть важное свойство!

Важное свойство

- Зафиксируем точку x_0
- В какую сторону функция быстрее всего растёт?

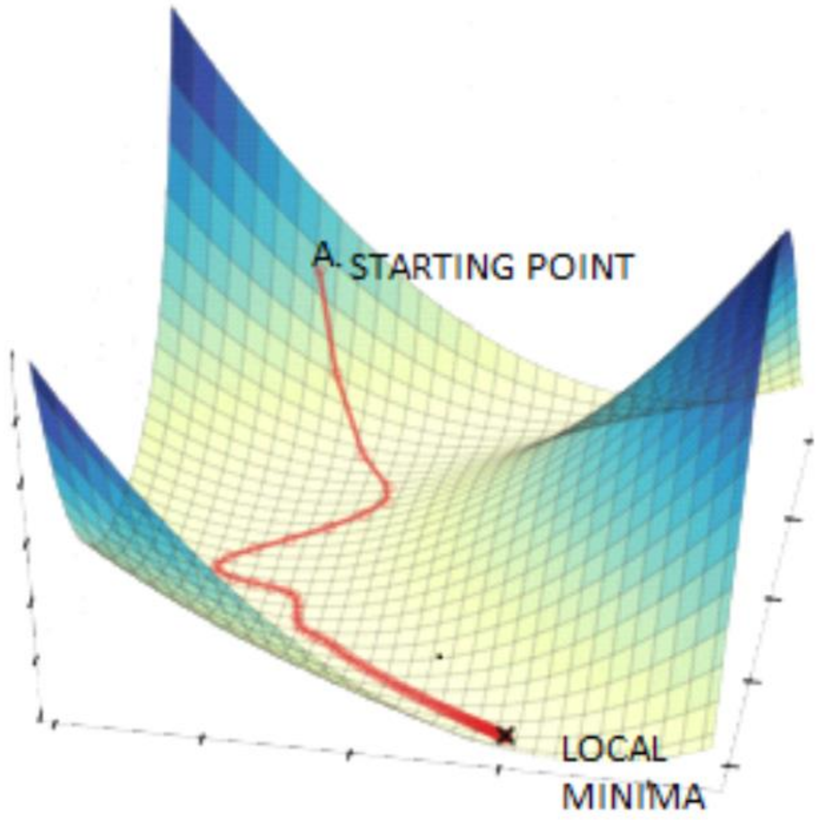


Важное свойство

- Зафиксируем точку x_0
- В какую сторону функция быстрее всего растёт?
- В направлении градиента!
- А быстрее всего убывает в сторону антиградиента

Как это пригодится?

Как это пригодится?



Градиентный спуск

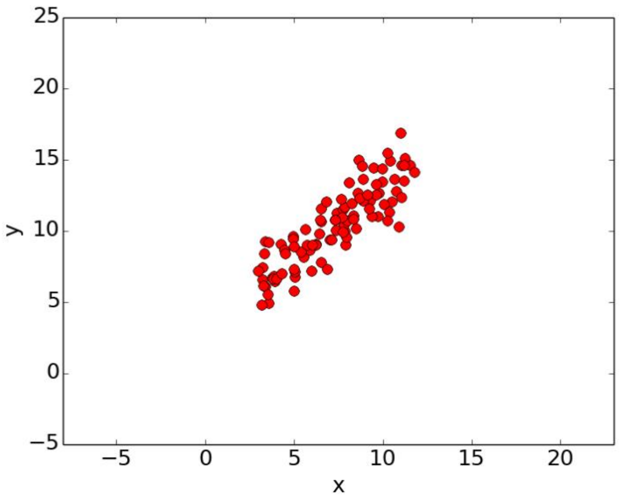
- Стартуем из случайной точки
- Сдвигаемся по антиградиенту
- Повторяем, пока не окажемся в точке минимума

Парная регрессия

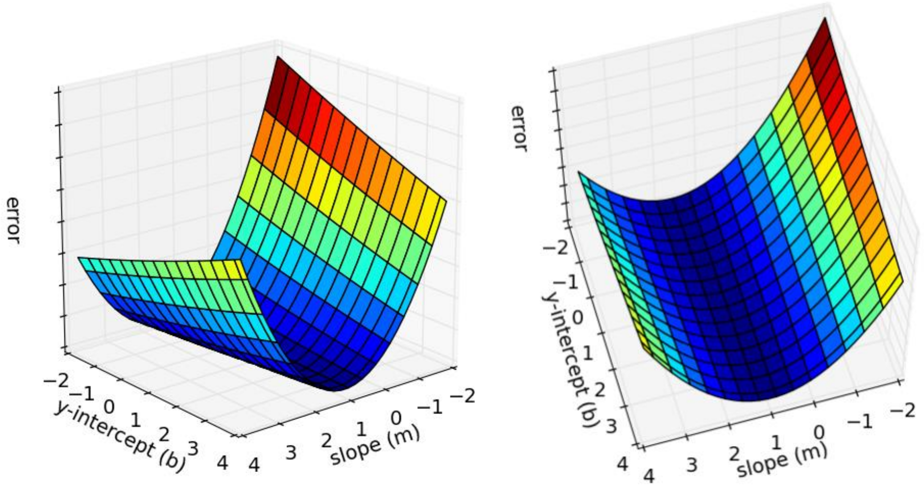
- Простейший случай: один признак
- Модель: $a(x) = w_1 x + w_0$
- Два параметра: w_1 и w_0
- Функционал:

$$Q(w_0, w_1) = \frac{1}{\ell} \sum_{i=1}^{\ell} (w_1 x_i + w_0 - y_i)^2$$

Парная регрессия



Выборка



Функционал ошибки

https://spin.atomicobject.com/2014/06/24/gradient-descent-linear-regression/

Начальное приближение

- w^0 — инициализация весов
- Например, из стандартного нормального распределения

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

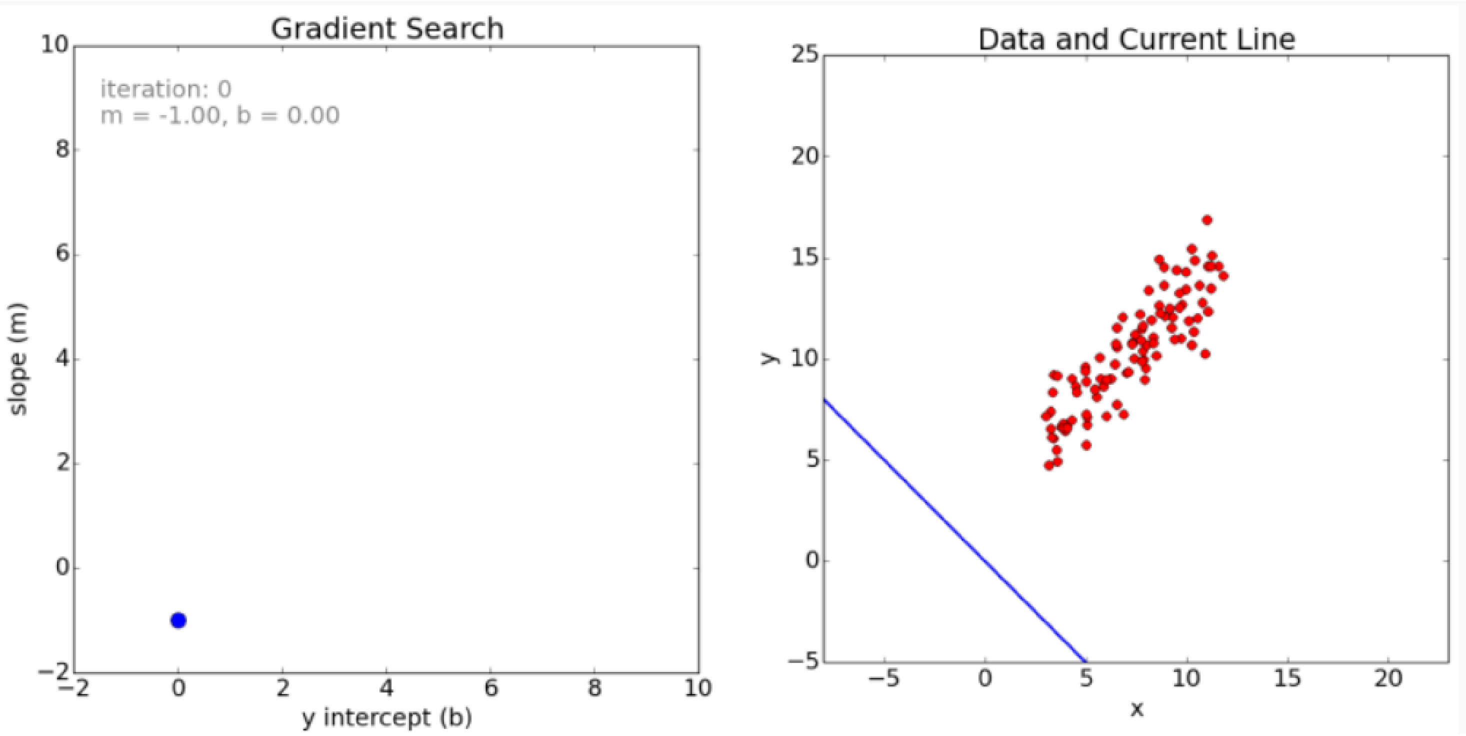
$$\|w^t - w^{t-1}\| < \varepsilon$$

- Другой вариант:

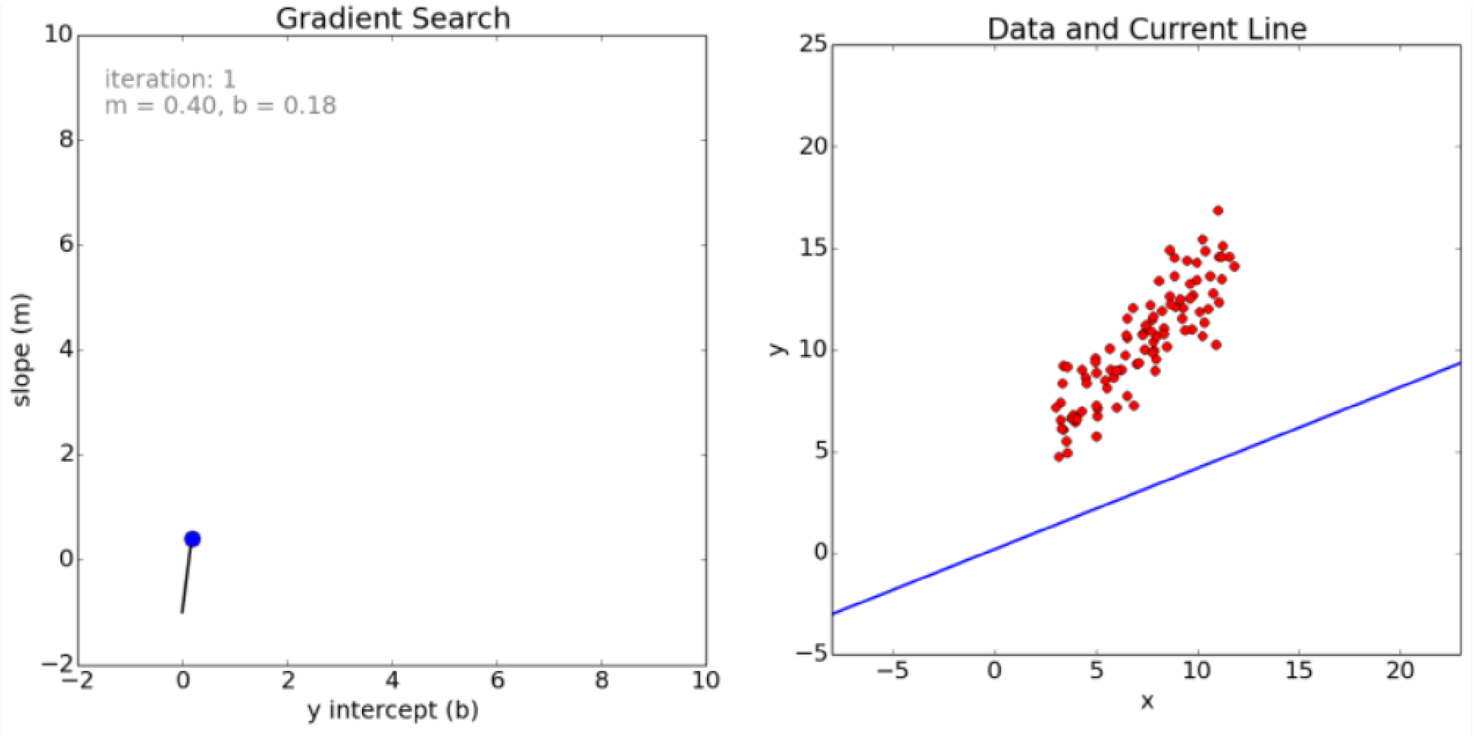
$$\|\nabla Q(w^t)\| < \varepsilon$$

- Или пока ошибка на отложенной выборке уменьшается

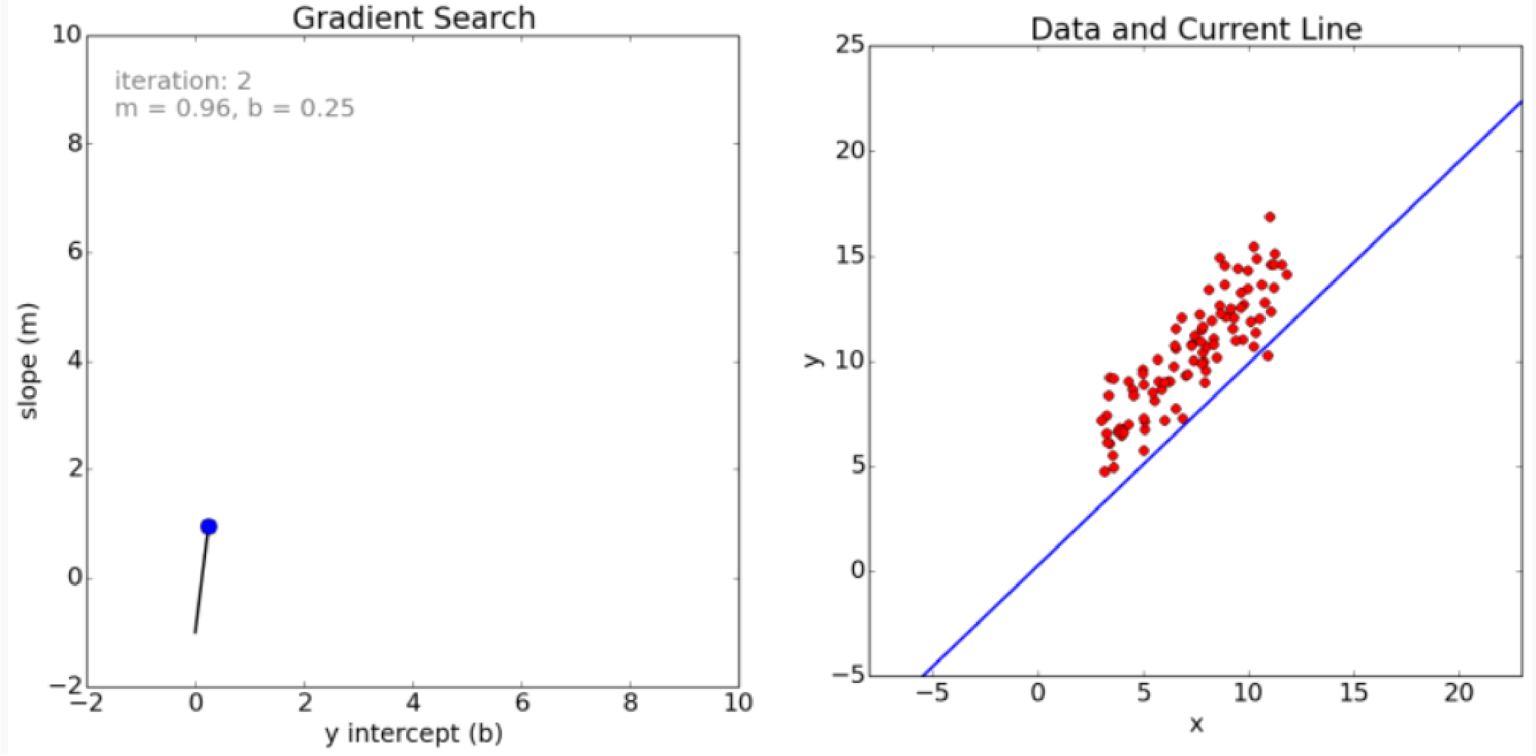
Парная регрессия



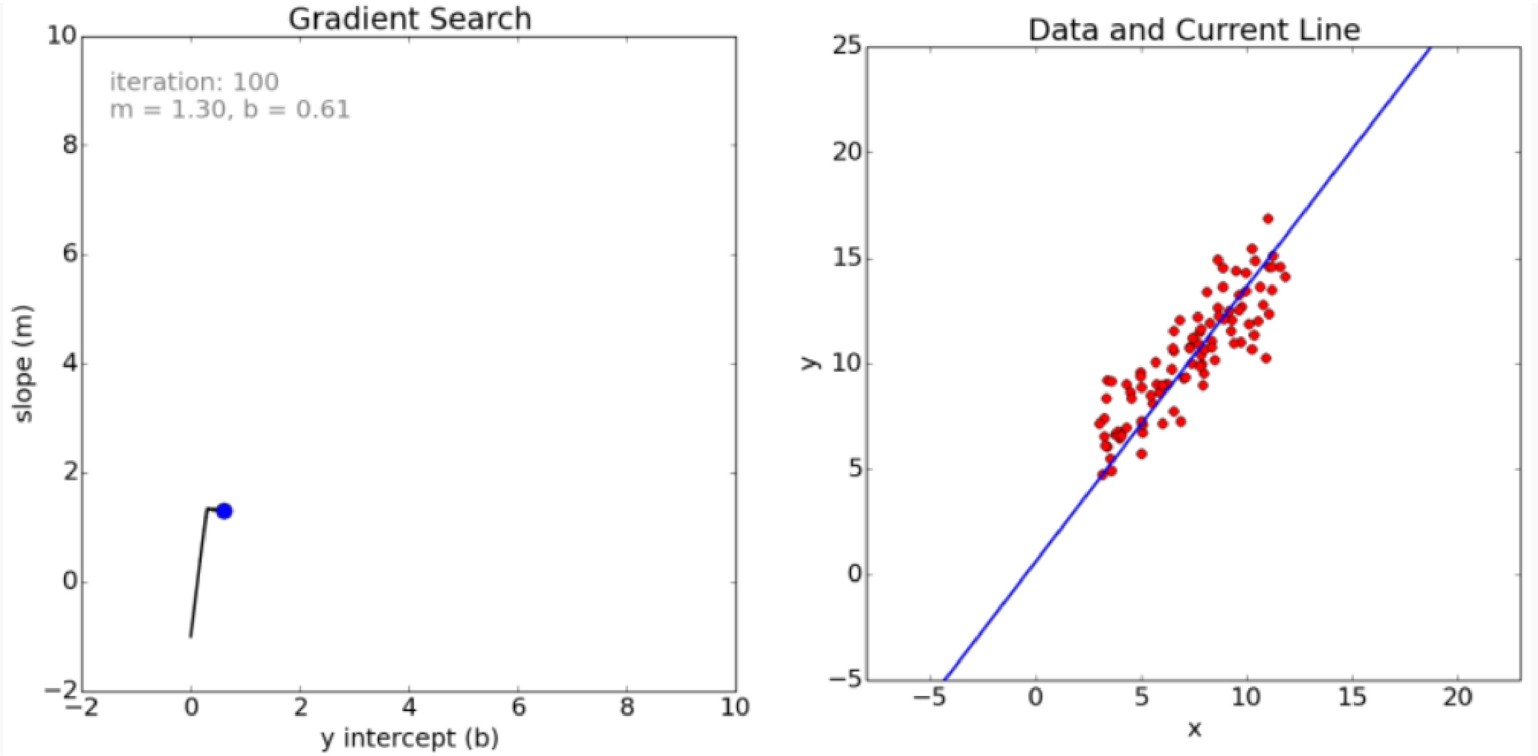
Парная регрессия



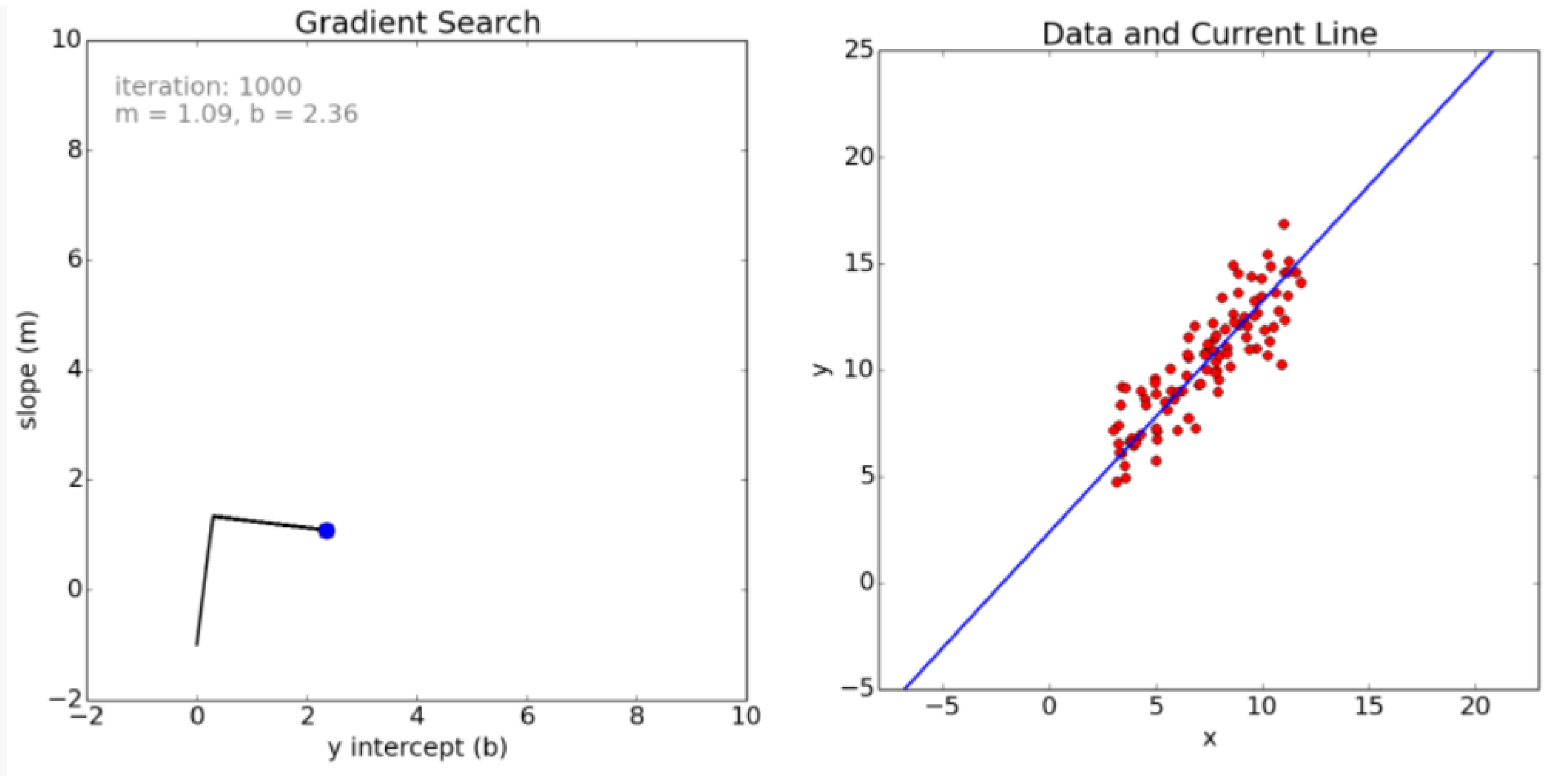
Парная регрессия



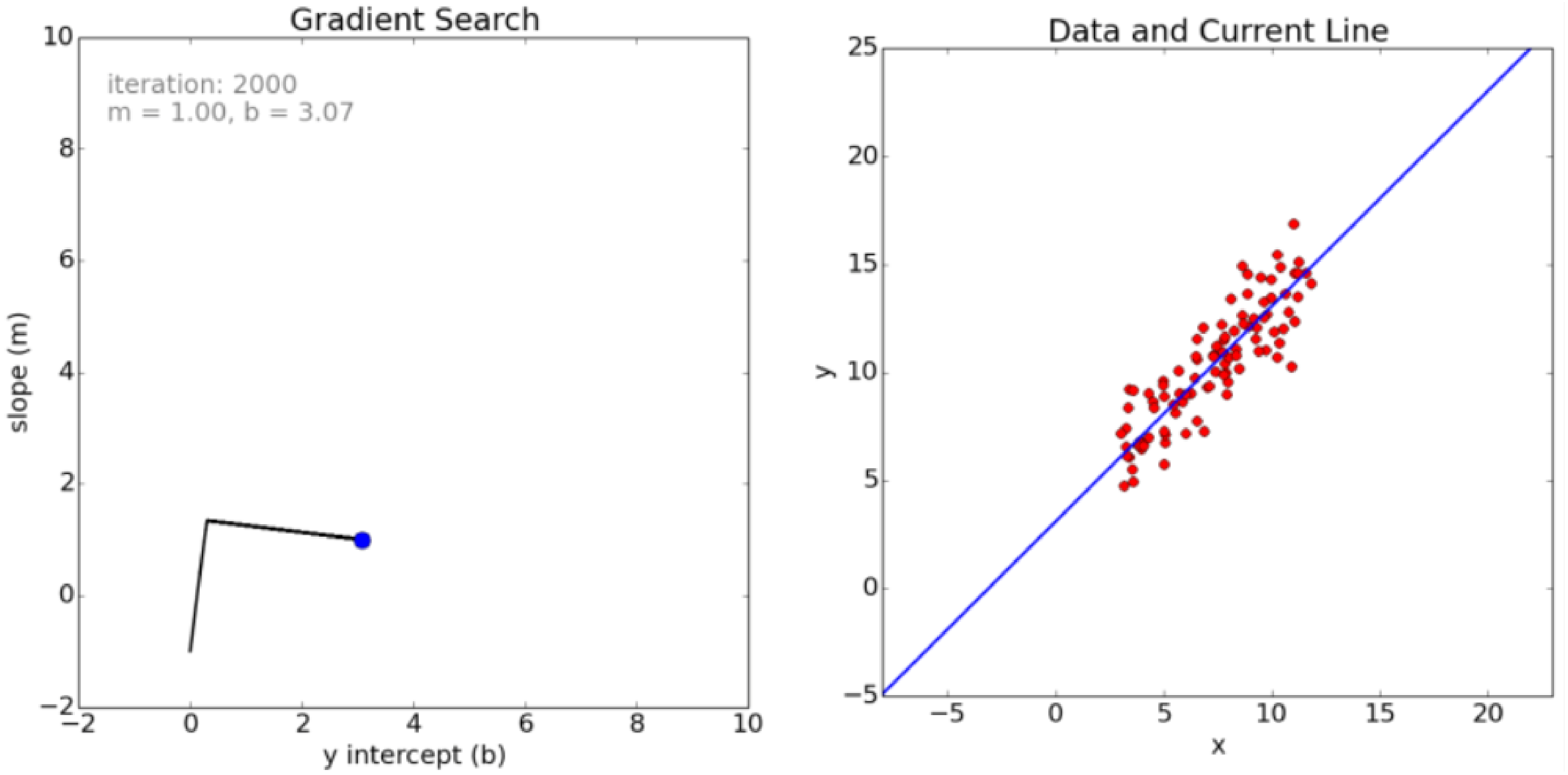
Парная регрессия



Парная регрессия

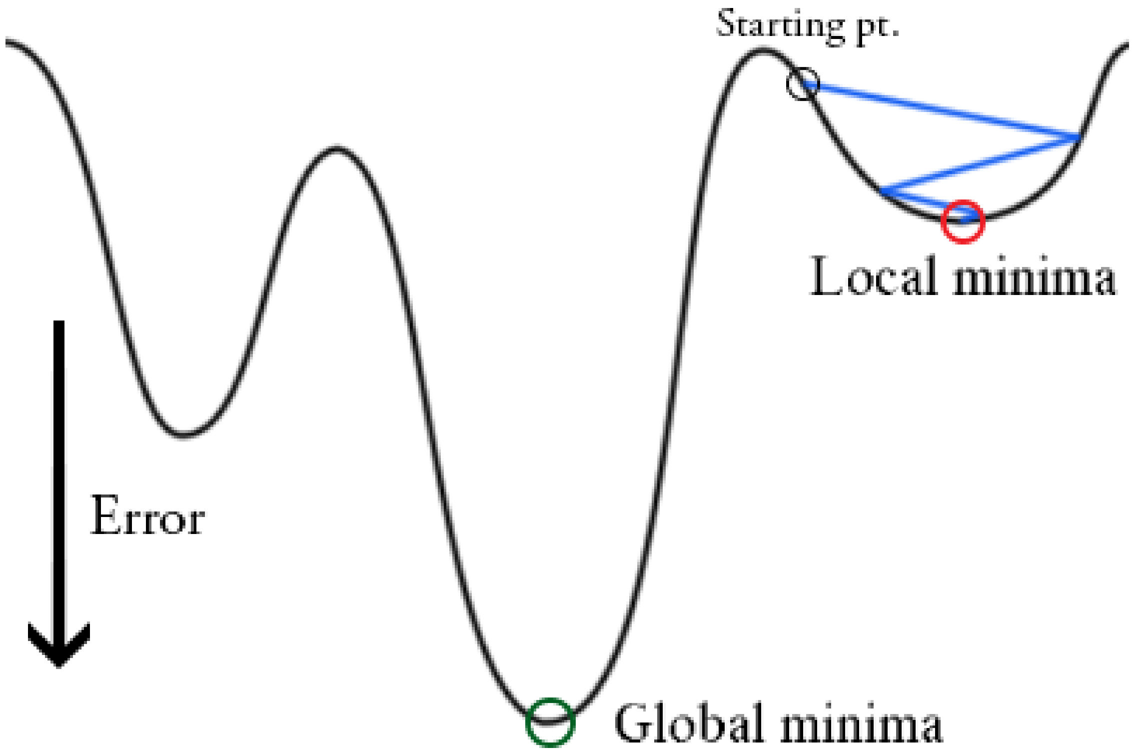


Парная регрессия

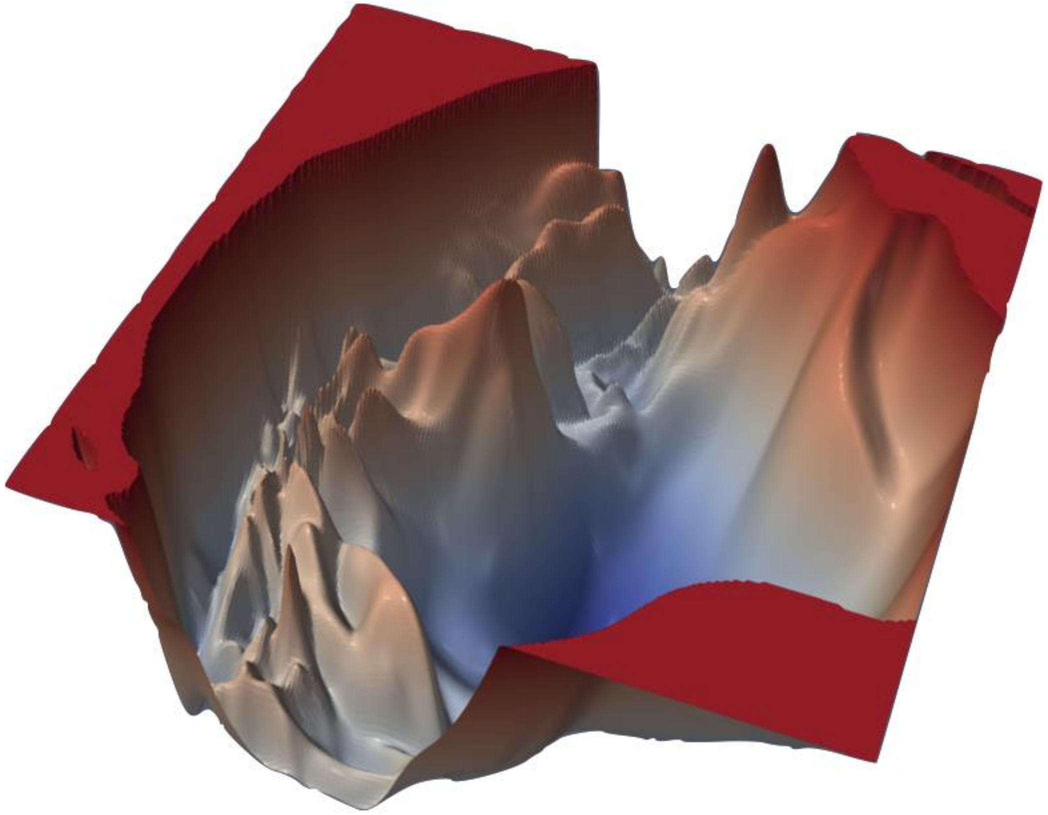


Локальные минимумы

- Градиентный спуск находит только локальные минимумы



Локальные минимумы



Резюме

- Градиент помогает понять, в какую сторону функция меняется быстрее всего
- Движение по антиградиенту позволяет найти локальный минимум

Глубинное обучение

Playground Prediction Competition

Dogs vs. Cats

Create an algorithm to distinguish dogs from cats

Kaggle · 213 teams · 7 years ago

Overview

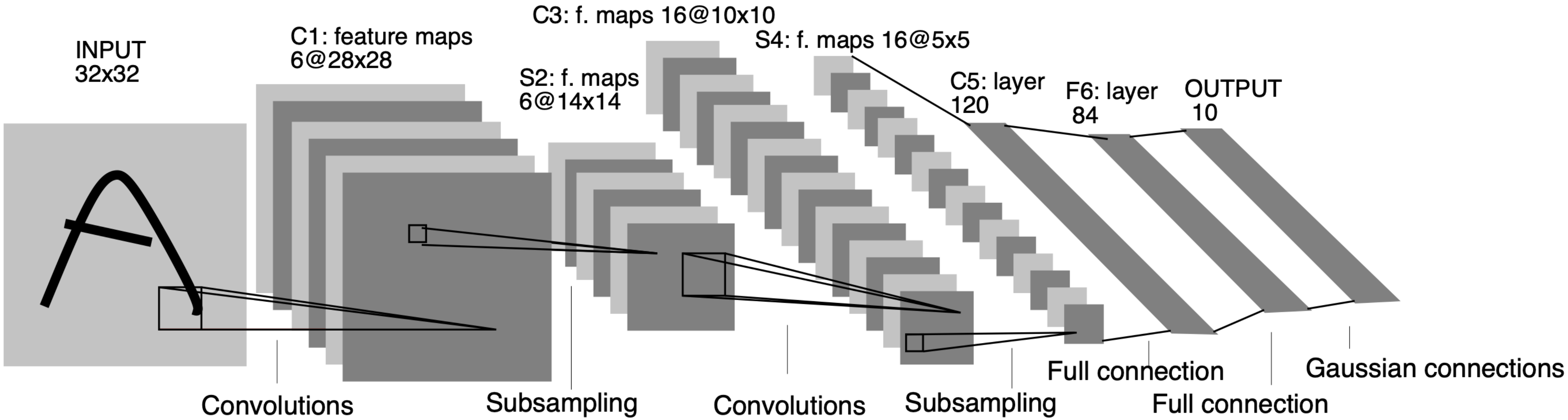
In this competition, you'll write an algorithm to classify whether images contain either a dog or a cat. This is easy for humans, dogs, and cats. Your computer will find it a bit more difficult.

Классическое компьютерное зрение

1. Считаем признаки (есть ли усы, какой формы уши, какой длины хвост, …)
2. Обучаем на них градиентный бустинг

- Посчитать признаки — целая история

Современное компьютерное зрение



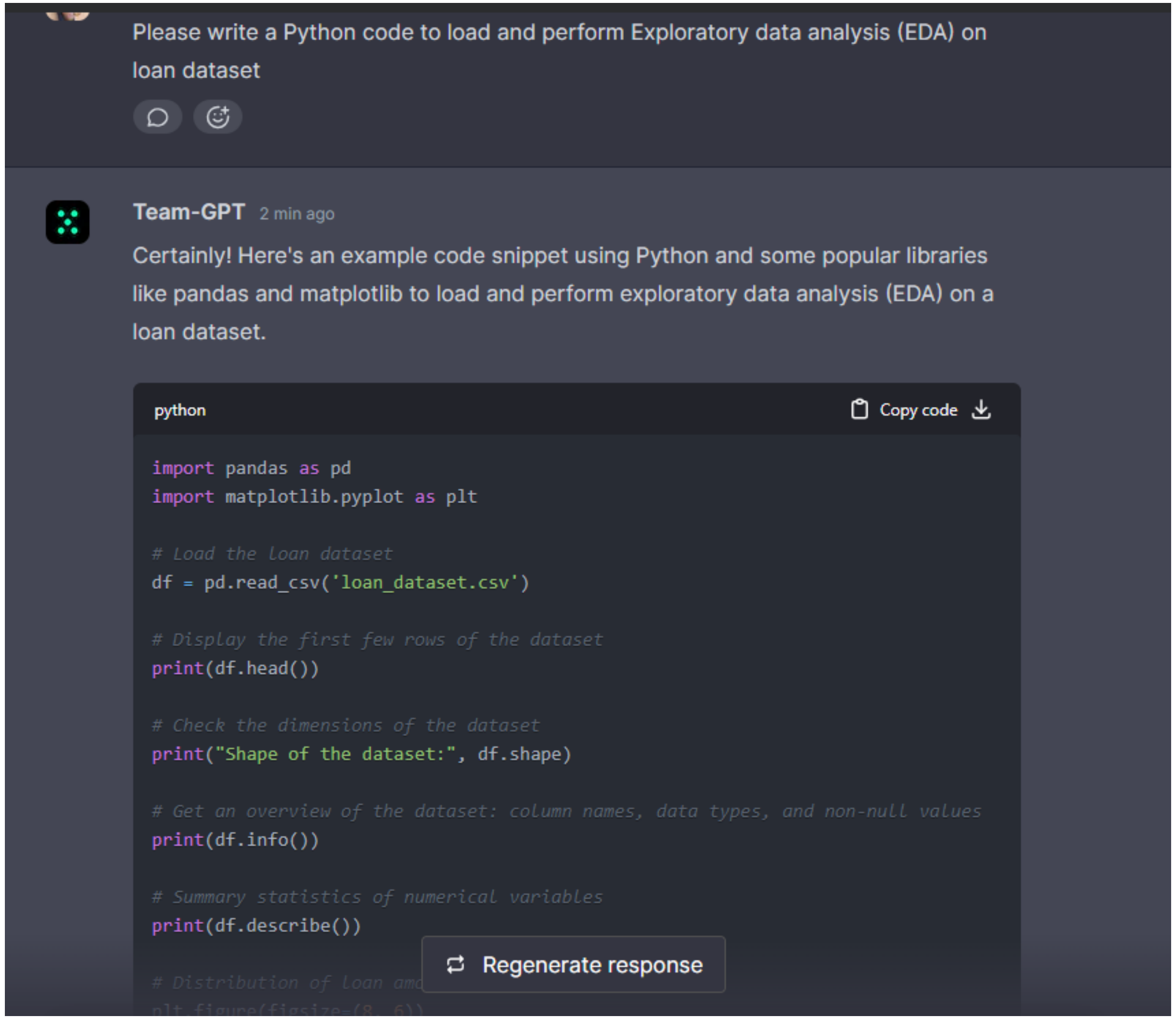
Классическое NLP

1. Подсчитываем статистику, как часто то или иное слово встречается после данного
2. Генерируем следующее слово из этого распределения

```
"Manure, almond gelato and frozen pies, you are also had it was in one but it will post office buildings s
ucks). their chinese food. comfort food while they liked their lids ripped off. it an early morning of jon
still a spade so maybe too much. the same. but, at the baked rigatoni, and not in other options and it see
ms odd taste). our visit). i go to nfl kickoff arrived with $. that's about when you come down hoyt street
is actually higher than impressed with a regular theater! so at it, halfway through their pork and though
i've"
```

https://medium.com/analytics-vidhya/making-a-text-generator-using-markov-chains-e17a67225d10

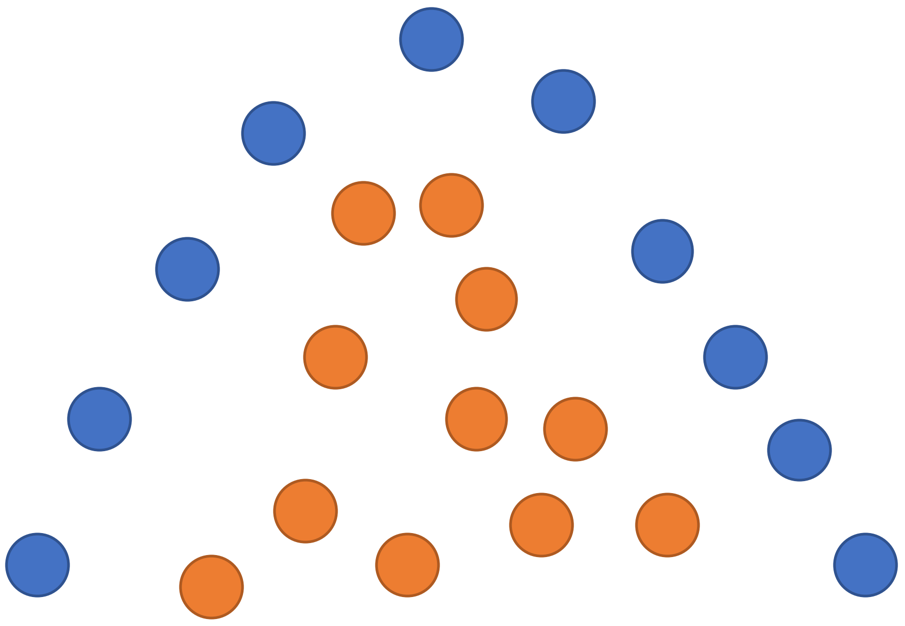
Современное NLP



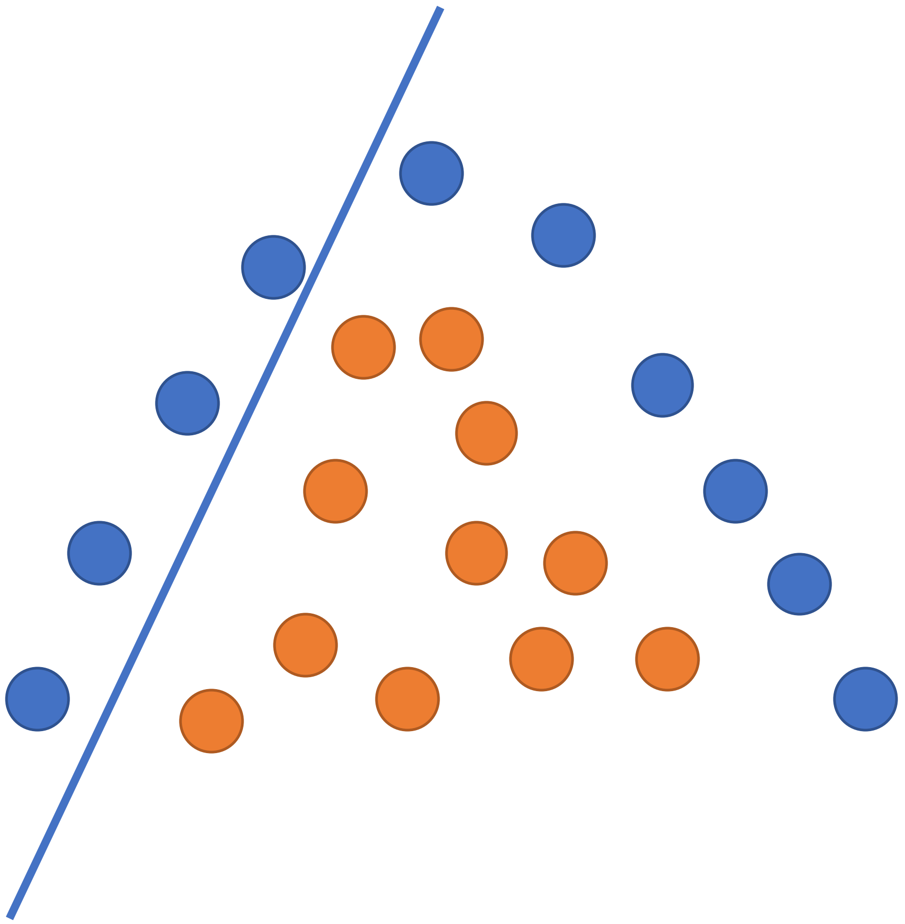
Успехи в глубинном обучении

- Изображения и видео
- Трёхмерное компьютерное зрение
- Тексты
- Звук
- Генерация данных

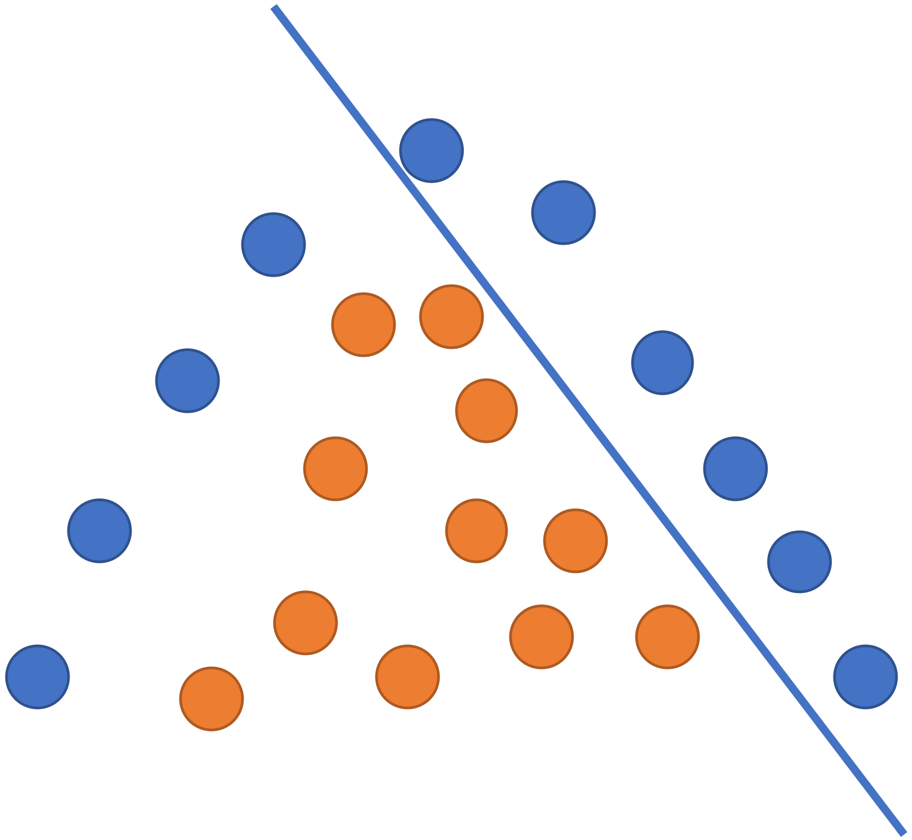
Нелинейные закономерности



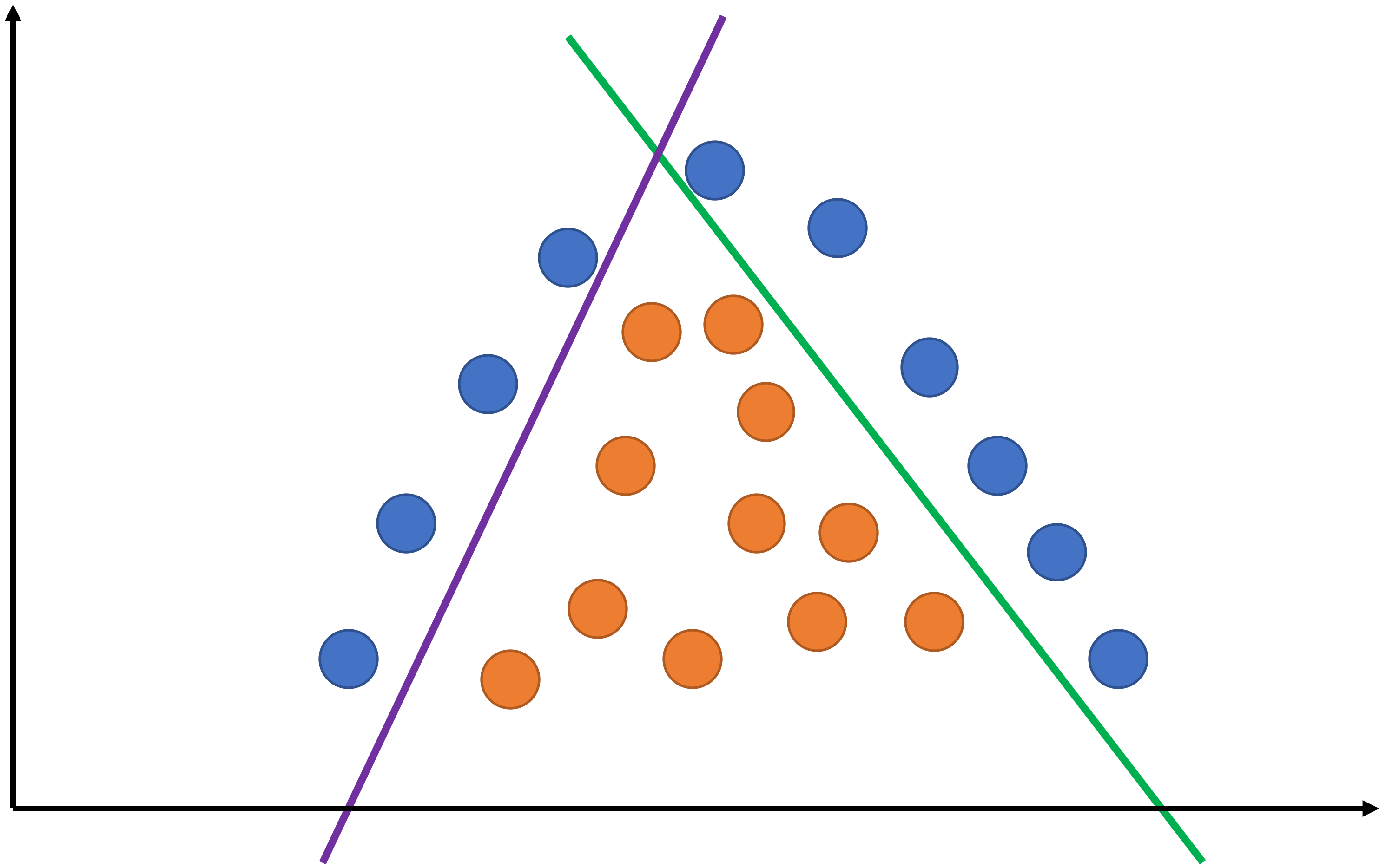
Нелинейные закономерности



Нелинейные закономерности



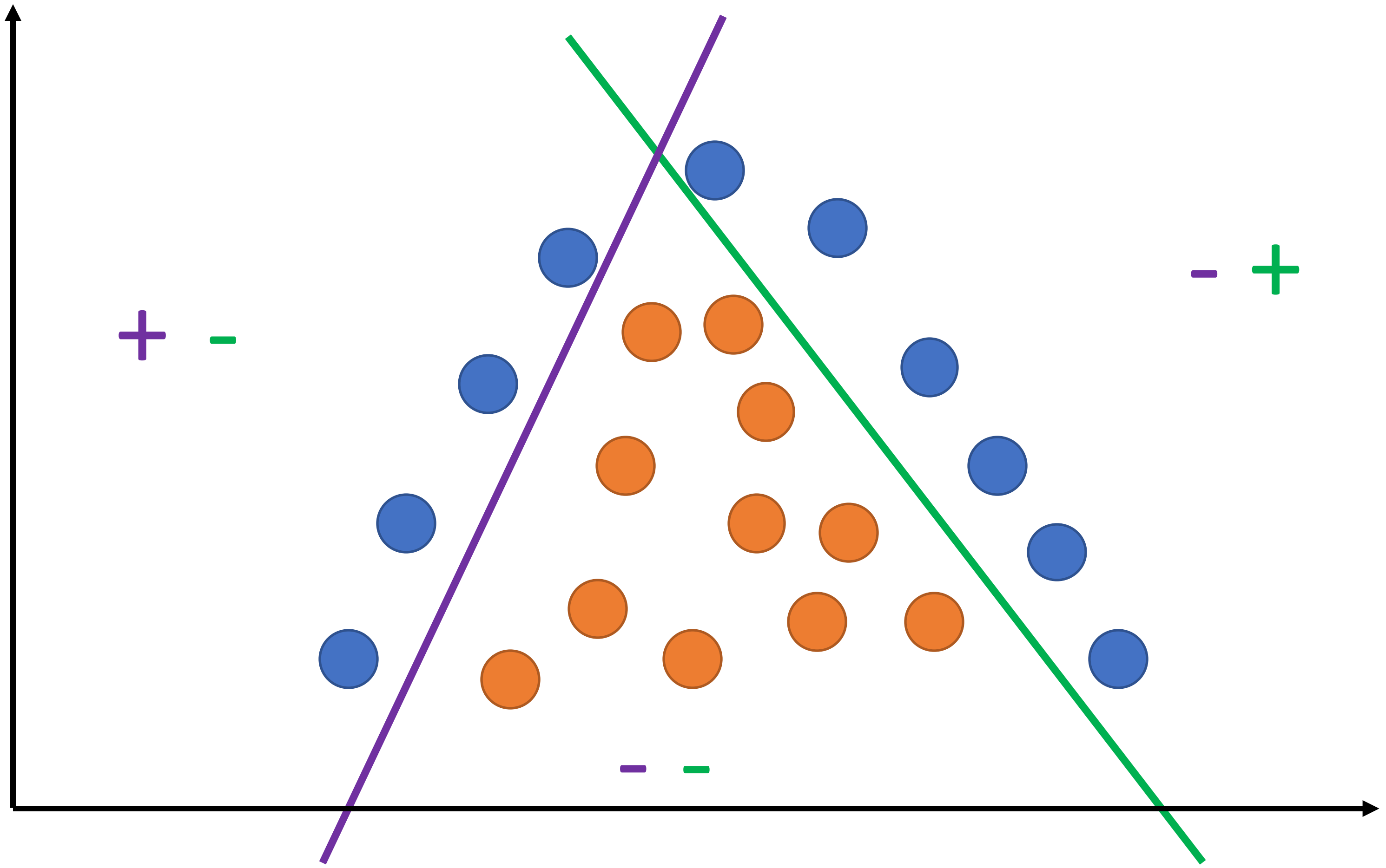
Нелинейные закономерности



$$b_1(x) = \mathrm{sign}(-2x_1 + x_2 + 5)$$

$$b_2(x) = \mathrm{sign}(2x_1 + x_2 - 5)$$

Нелинейные закономерности

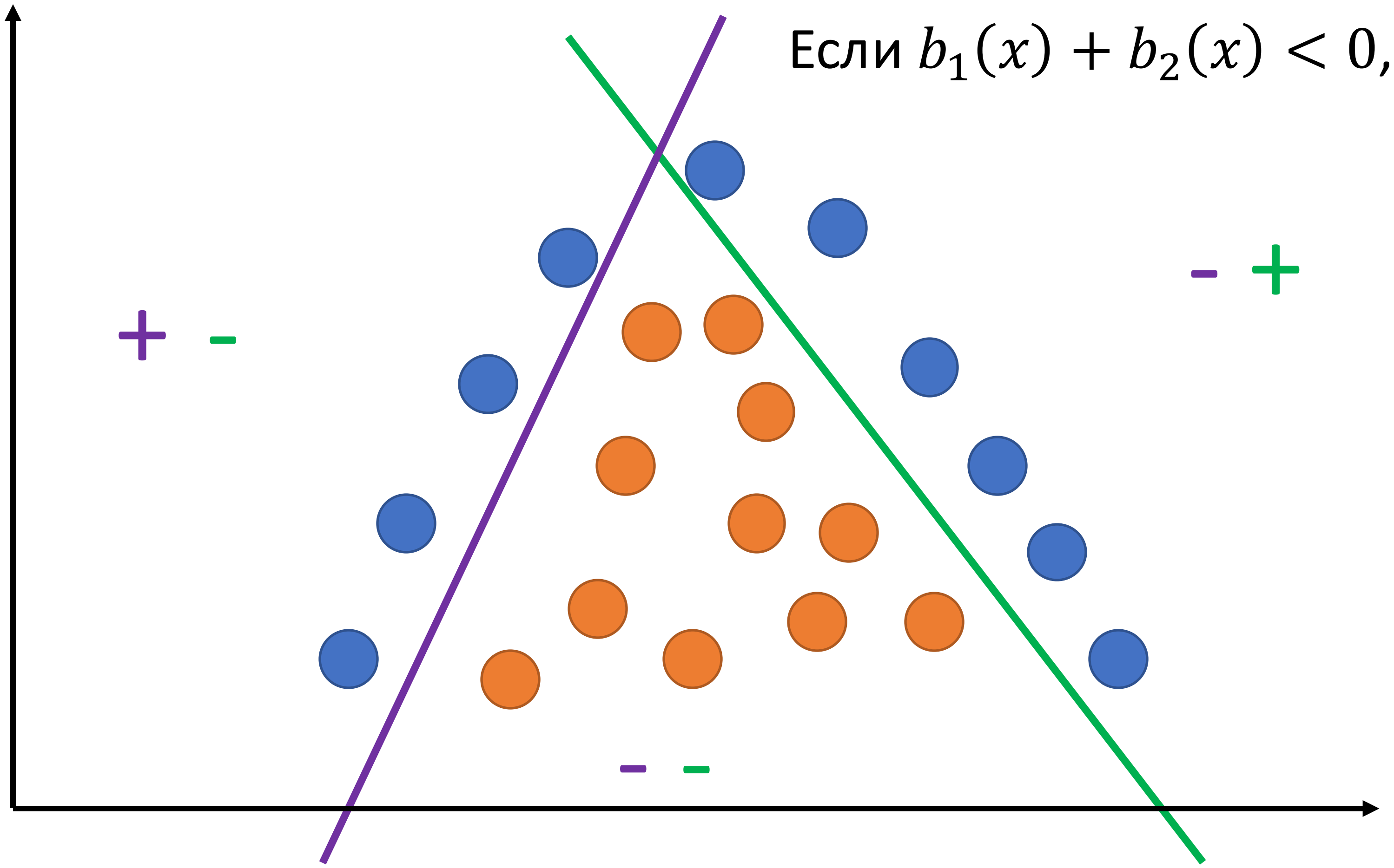


$$b_1(x) = \mathrm{sign}(-2x_1 + x_2 + 5)$$

$$b_2(x) = \mathrm{sign}(2x_1 + x_2 - 5)$$

Нелинейные закономерности

Если $b_1(x) + b_2(x) < 0$, то оранжевый класс



$b_1(x) = \mathrm{sign}(-2x_1 + x_2 + 5)$

$b_2(x) = \mathrm{sign}(2x_1 + x_2 - 5)$

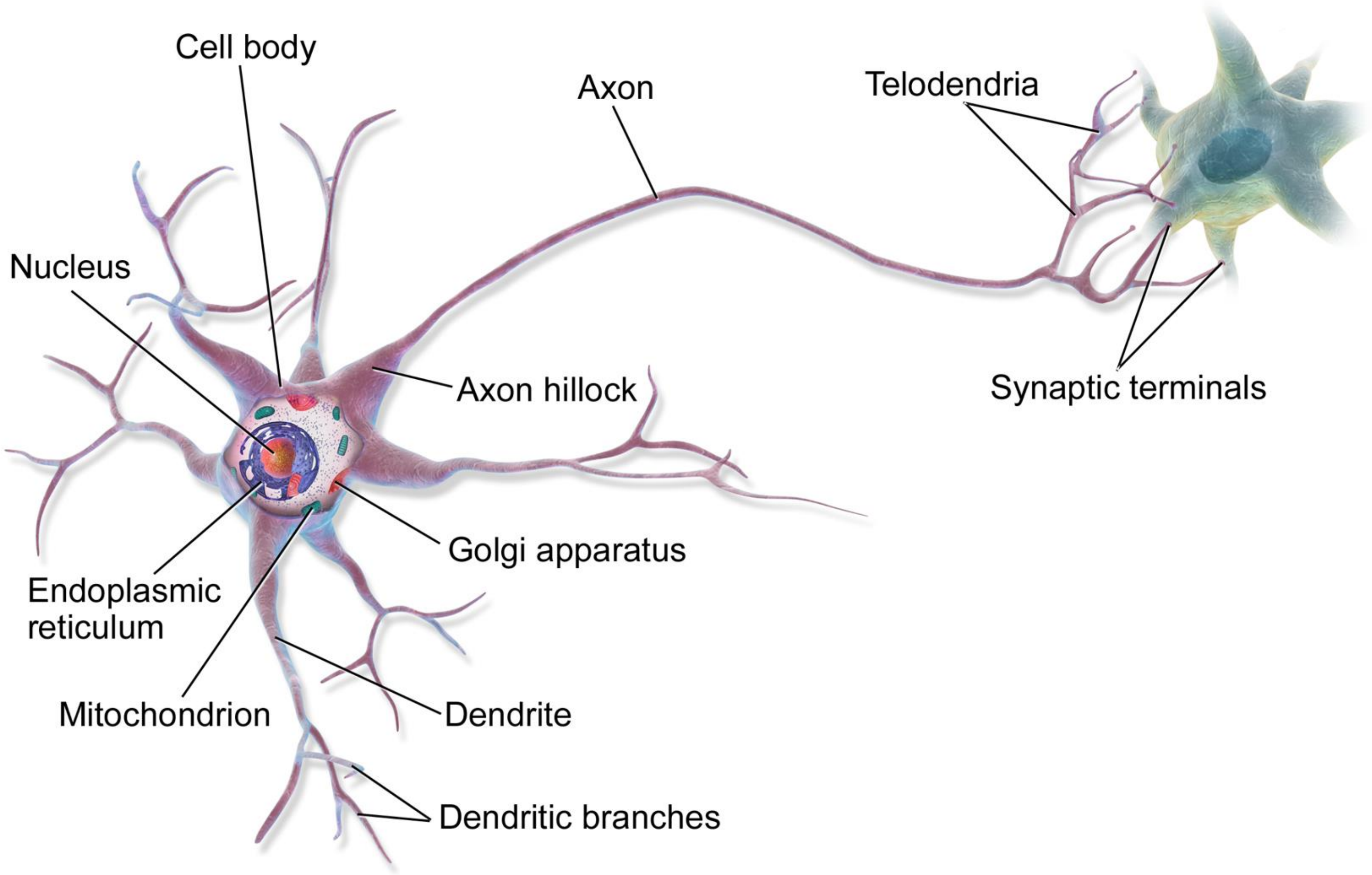
Нелинейные закономерности

$$z_1 = -2x_1 + x_2 + 5 \rightarrow b_1 = \operatorname{sign}(z_1)$$

$$z_2 = 2x_1 + x_2 - 5 \rightarrow b_2 = \operatorname{sign}(z_2)$$

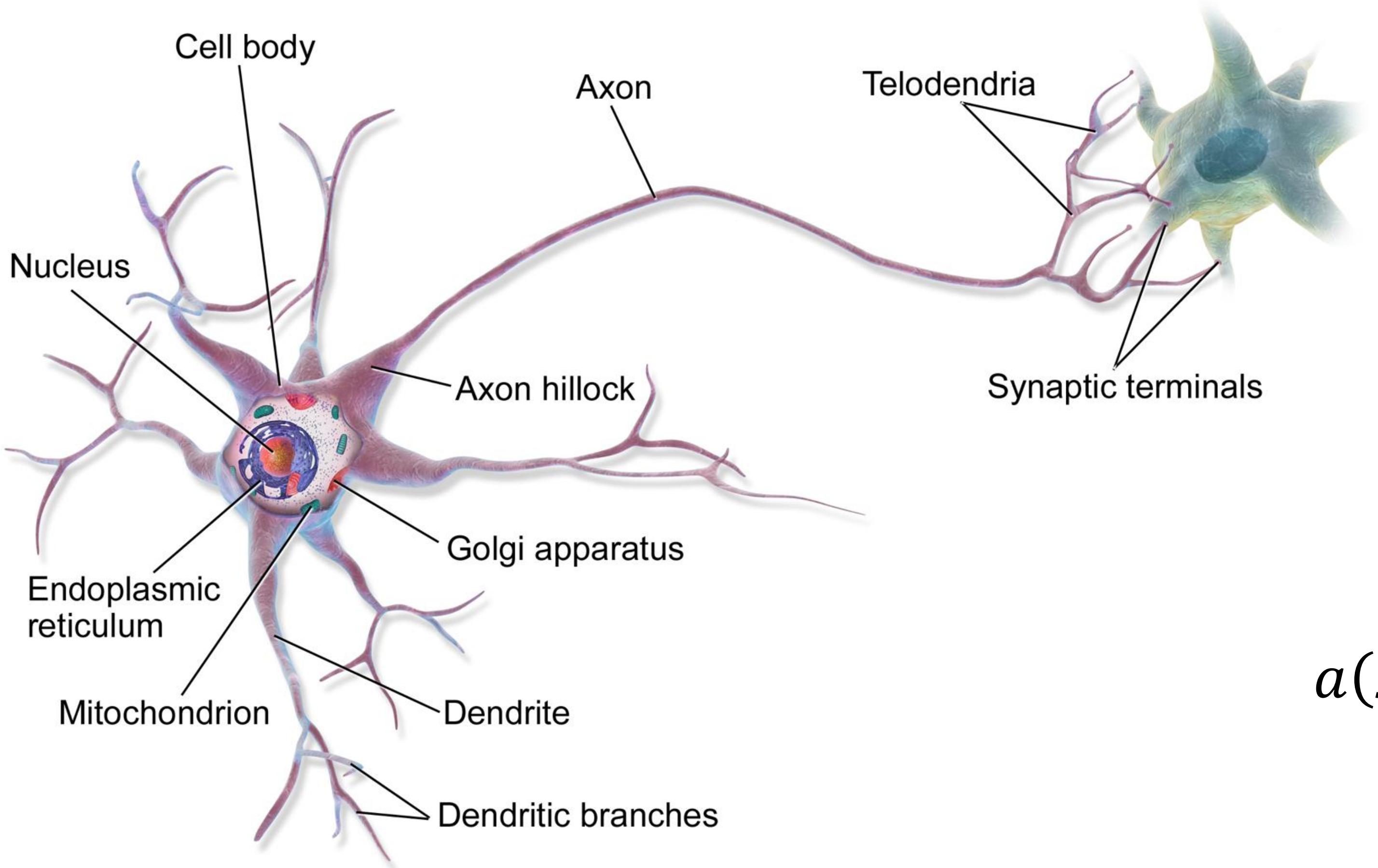
$$a = b_1 + b_2 \rightarrow [a < 0]$$

Нейрон



https://en.wikipedia.org/wiki/Neuron

Нейрон



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

https://en.wikipedia.org/wiki/Neuron

Нелинейные закономерности

как бы нейрон

$$
\begin{tikzcd}
z_1 = -2x_1 + x_2 + 5 \arrow[r] & b_1 = \mathrm{sign}(z_1) \arrow[dr] & & \\
 & & a = b_1 + b_2 \arrow[r] & {[a < 0]} \\
z_2 = 2x_1 + x_2 - 5 \arrow[r] & b_2 = \mathrm{sign}(z_2) \arrow[ur] & &
\end{tikzcd}
$$

Нелинейные закономерности

как бы нейронная сеть

$$
\begin{tikzcd}
\boxed{z_1 = -2x_1 + x_2 + 5} \arrow[r] & \boxed{b_1 = \operatorname{sign}(z_1)} \arrow[dr] & & \\
 & & \boxed{a = b_1 + b_2} \arrow[r] & \boxed{[a < 0]} \\
\boxed{z_2 = 2x_1 + x_2 - 5} \arrow[r] & \boxed{b_2 = \operatorname{sign}(z_2)} \arrow[ur] & &
\end{tikzcd}
$$

Граф вычислений (или нейронная сеть)

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

$$\boxed{x^{(0)}} \longrightarrow \boxed{h_1} \xrightarrow{x^{(1)}}$$

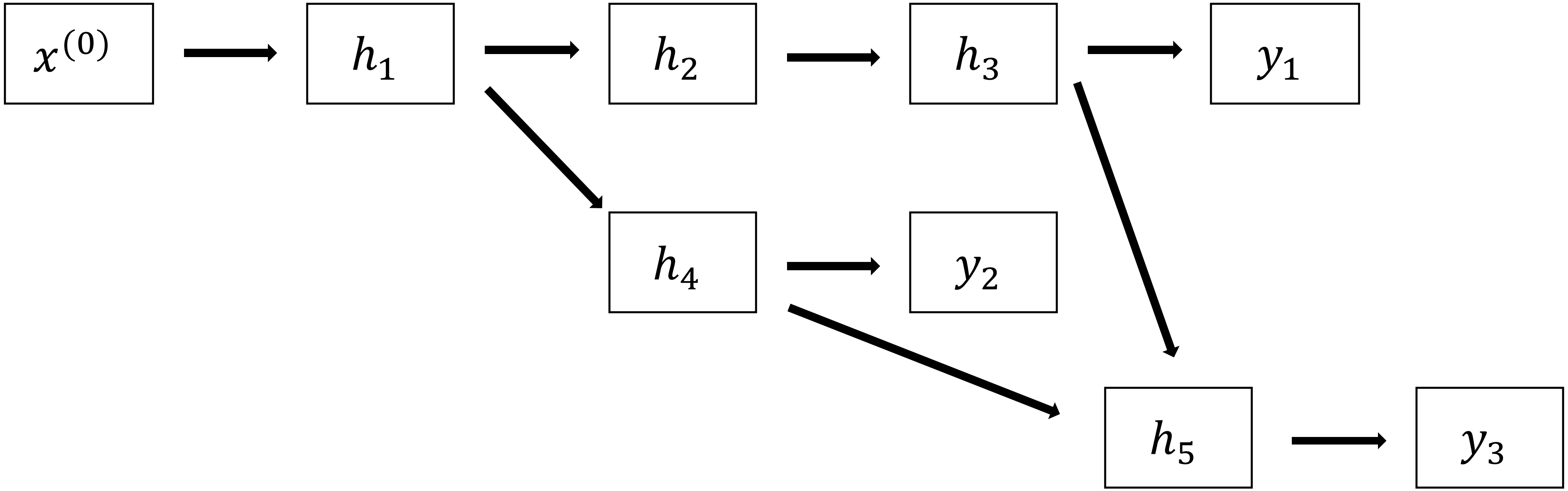
Граф вычислений (или нейронная сеть)

$$\boxed{x^{(0)}} \longrightarrow \boxed{h_1} \xrightarrow{x^{(1)}} \boxed{h_2} \xrightarrow{x^{(2)}} \dots$$

Граф вычислений (или нейронная сеть)

$$\boxed{x^{(0)}} \longrightarrow \boxed{h_1} \xrightarrow{x^{(1)}} \boxed{h_2} \xrightarrow{x^{(2)}} \boxed{h_3} \longrightarrow \boxed{y}$$

Граф вычислений (или нейронная сеть)



Резюме

- Нейронные сети — модели, состоящие из большого количества слоёв
- Каждый слой — модель, извлекающая новые признаки (в некотором смысле)

Полносвязные слои

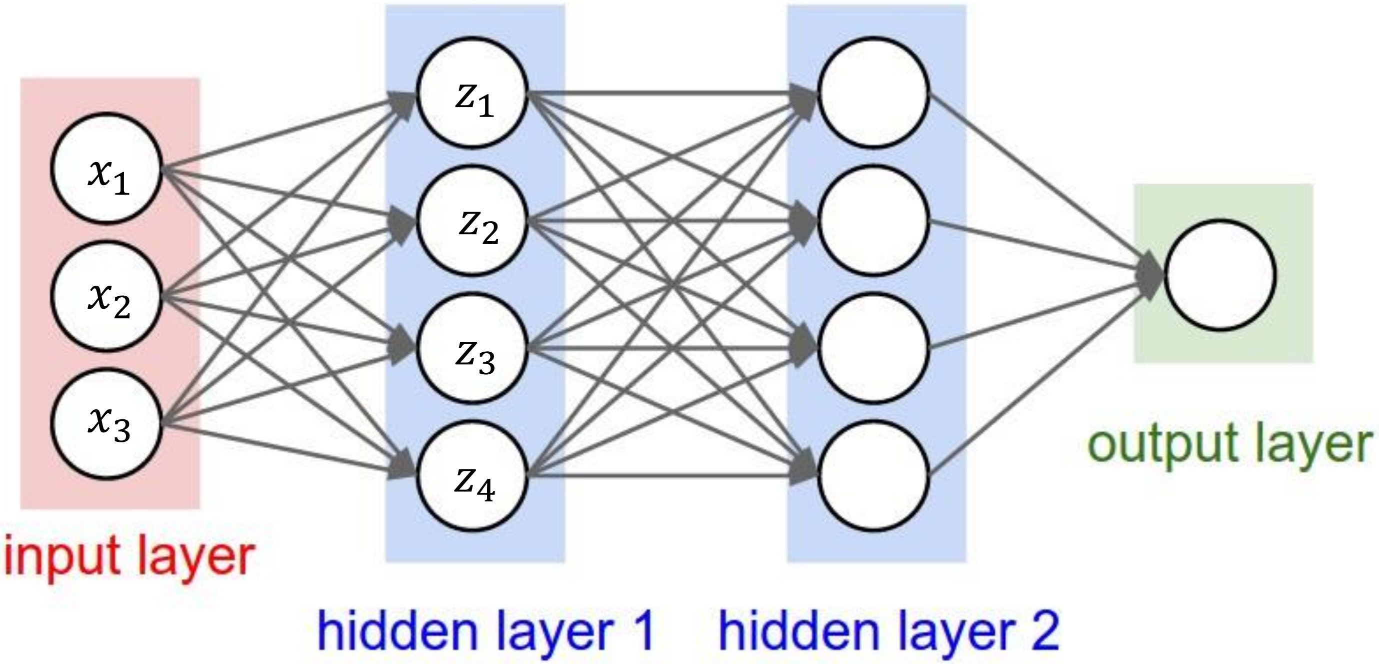
Полносвязный слой (fully connected, FC)

- 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)

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



https://cs231n.github.io/convolutional-networks/

Полносвязный слой (fully connected, FC)

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

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

Полносвязный слой (fully connected, FC)

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

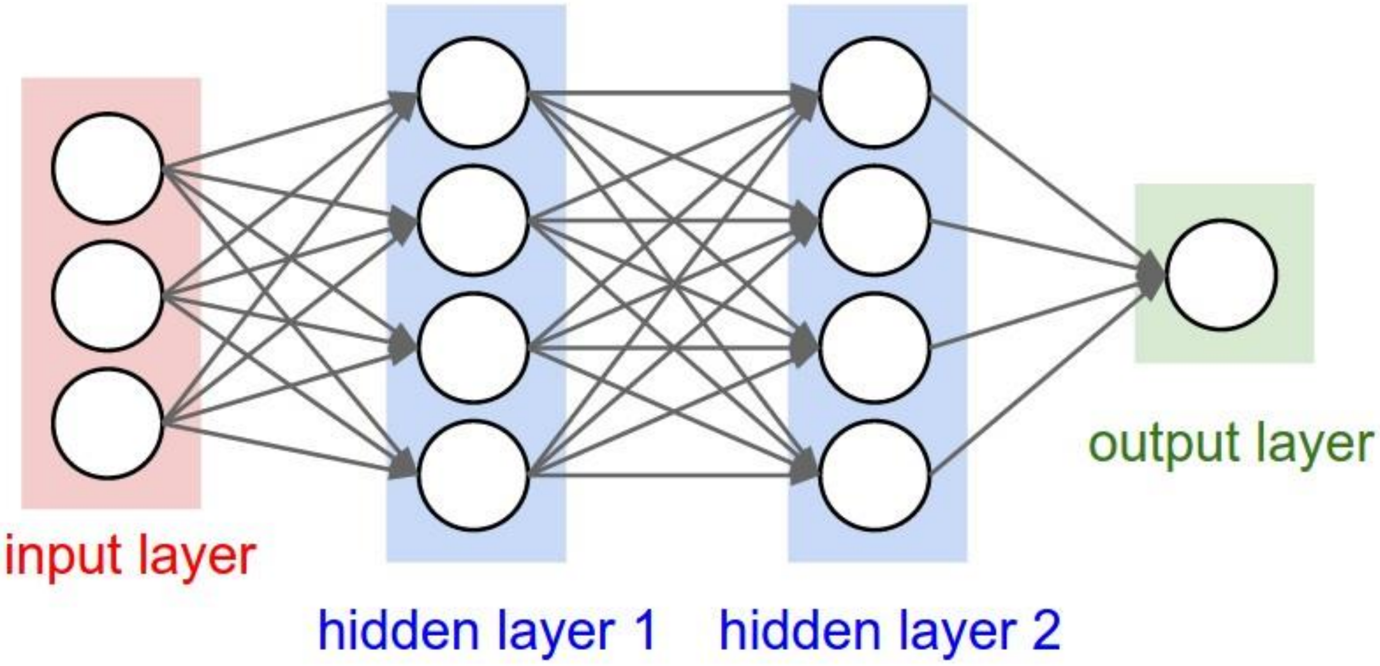
- m линейных моделей, в каждой $(n + 1)$ параметров
- Всего примерно mn параметров в полносвязном слое
- Это очень много: если у нас 1.000.000 входных признаков и 1000 выходов, то это 1.000.000.000 параметров

Важный вопрос в DL

Как объединить слои в мощную модель?

Нелинейность

- Рассмотрим два полносвязных слоя



Нелинейность

- Рассмотрим два полносвязных слоя

$$s_k = \sum_{j=1}^{m} v_{kj} z_j + c_k = \sum_{j=1}^{m} v_{kj} \sum_{i=1}^{n} w_{ji} x_i + \sum_{j=1}^{m} v_{kj} b_j + c_k =$$

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

- То есть это ничем не лучше одного полносвязного слоя

Нелинейность

- Нужно добавлять нелинейную функцию после полносвязного слоя

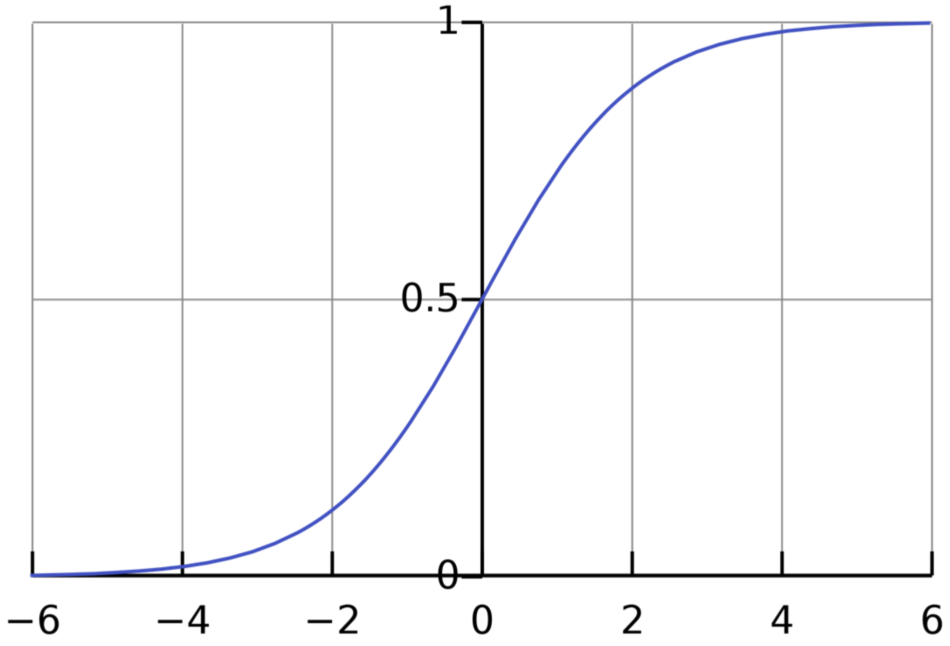
$$z_j = f\left(\sum_{i=1}^{n} w_{ji} x_i + b_j\right)$$

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)

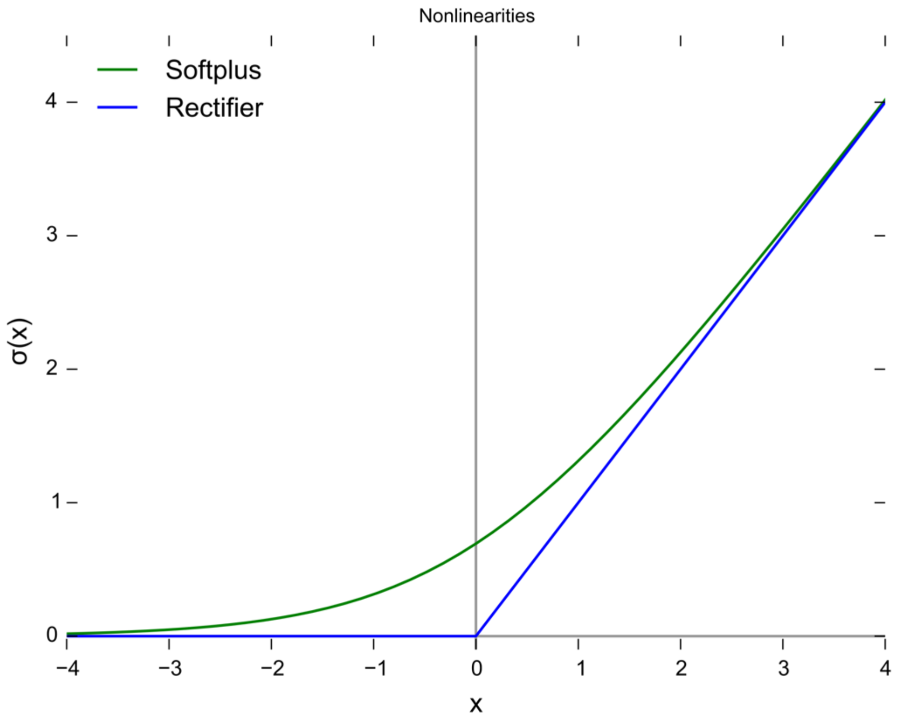


Нелинейность

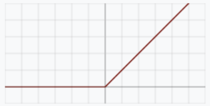
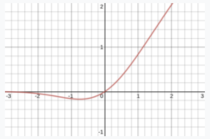
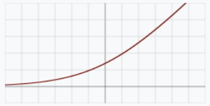
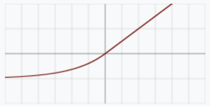
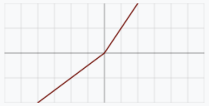
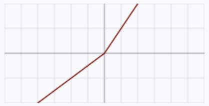
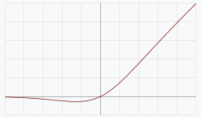
$$z_j = f\left(\sum_{i=1}^{n} w_{ji}x_i + b_j\right)$$

Вариант 2: $f(x) = \max(0, x)$

(ReLU, REctified Linear Unit)



Нелинейность

Rectified linear unit (ReLU)[9]		$\begin{cases} 0 & \text{if } x \le 0 \\ x & \text{if } x > 0 \end{cases}$ $= \max\{0, x\} = x\mathbf{1}_{x>0}$
Gaussian Error Linear Unit (GELU)[4]		$\frac{1}{2}x\left(1 + \text{erf}\left(\frac{x}{\sqrt{2}}\right)\right)$ $= x\Phi(x)$
Softplus[10]		$\ln(1 + e^x)$
Exponential linear unit (ELU)[11]		$\begin{cases} \alpha\left(e^x - 1\right) & \text{if } x \le 0 \\ x & \text{if } x > 0 \end{cases}$ with parameter α
Scaled exponential linear unit (SELU)[12]		$\lambda\begin{cases} \alpha(e^x - 1) & \text{if } x < 0 \\ x & \text{if } x \ge 0 \end{cases}$ with parameters $\lambda = 1.0507$ and $\alpha = 1.67326$
Leaky rectified linear unit (Leaky ReLU)[13]		$\begin{cases} 0.01x & \text{if } x < 0 \\ x & \text{if } x \ge 0 \end{cases}$
Parameteric rectified linear unit (PReLU)[14]		$\begin{cases} \alpha x & \text{if } x < 0 \\ x & \text{if } x \ge 0 \end{cases}$ with parameter α
Sigmoid linear unit (SiLU,[4] Sigmoid shrinkage,[15] SiL,[16] or Swish-1[17])		$\frac{x}{1 + e^{-x}}$

https://en.wikipedia.org/wiki/Activation_function

Типичная полносвязная сеть

$$\boxed{x^{(0)}} \longrightarrow \boxed{\mathrm{FC}_1} \longrightarrow \boxed{f} \longrightarrow \boxed{\mathrm{FC}_2} \longrightarrow \boxed{f} \longrightarrow \dots$$

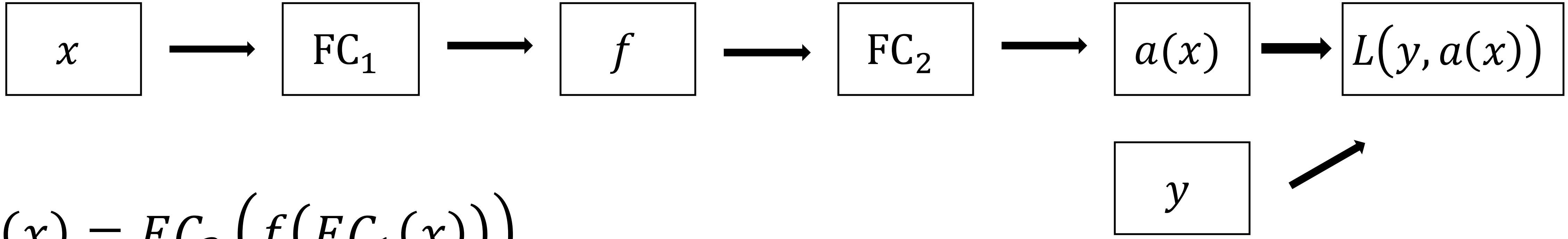
Типичная полносвязная сеть

$$x^{(0)} \longrightarrow \mathrm{FC}_1 \longrightarrow f \longrightarrow \mathrm{FC}_2 \longrightarrow f \longrightarrow \dots$$

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

Обучение нейронных сетей

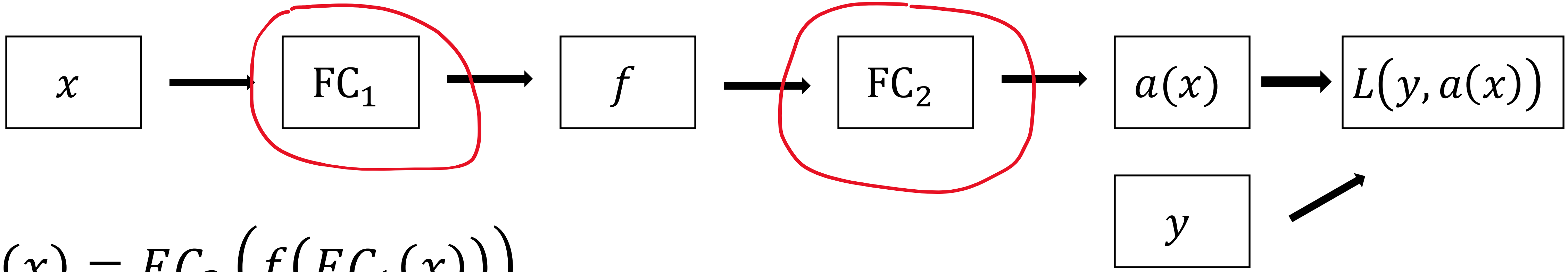
- Все слои обычно дифференцируемы, поэтому можно посчитать производные по всем параметрам



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

Обучение нейронных сетей

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



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

Обучение нейронных сетей

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

$$x \rightarrow \mathrm{FC}_1 \rightarrow f \rightarrow \mathrm{FC}_2 \rightarrow a(x) \rightarrow L\big(y, a(x)\big), \quad y \nearrow L\big(y, a(x)\big)$$

- $a(x) = FC_2\Big(f\big(FC_1(x)\big)\Big)$

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

Считаем производные

- Для градиентного спуска нужны производные ошибки по параметрам:

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

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?

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

Считаем производные

- Для градиентного спуска нужны производные ошибки по параметрам:

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

как сильно изменится ошибка, если пошевелить w_j?

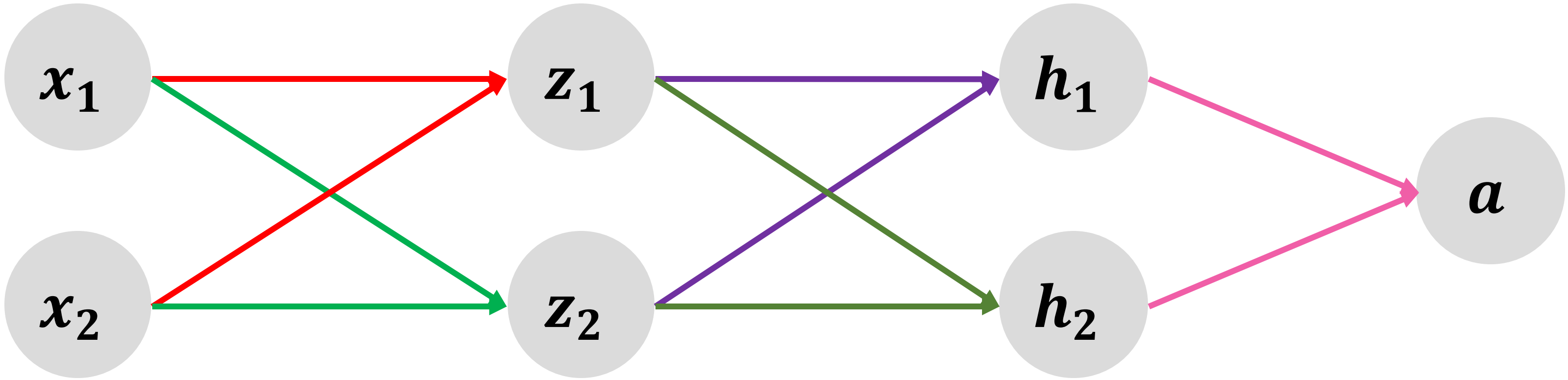
как сильно изменится ошибка, если пошевелить $a(x_i, w)$?

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

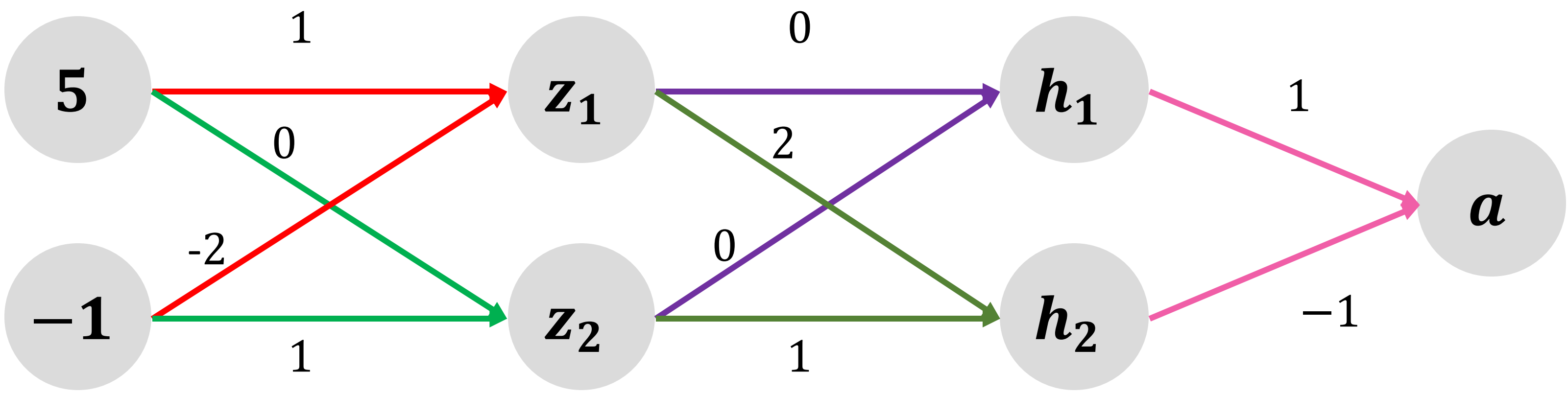
Считаем производные

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

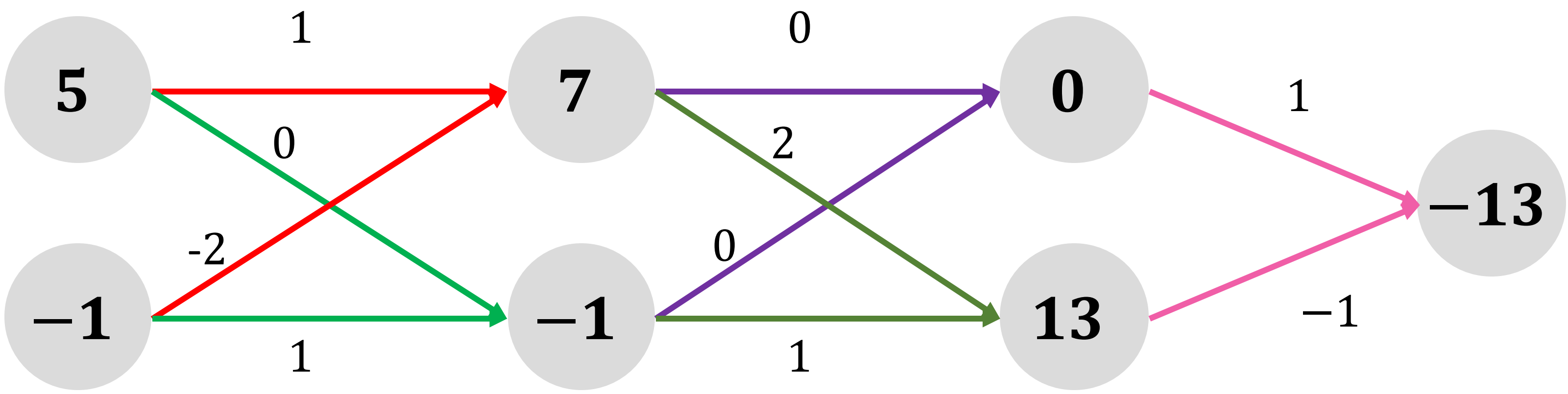
Как считать производные?



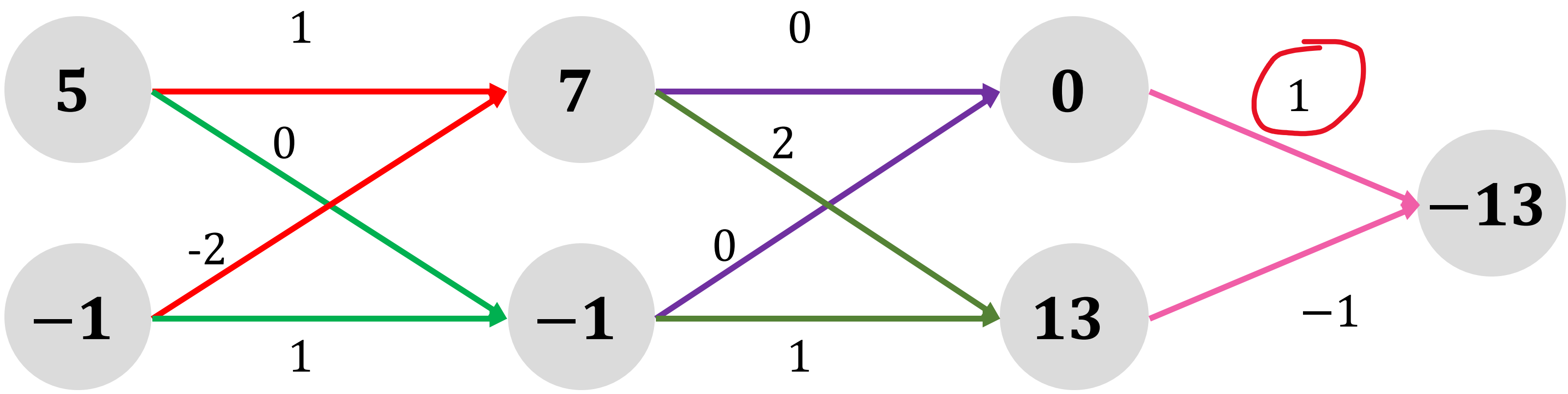
Как считать производные?



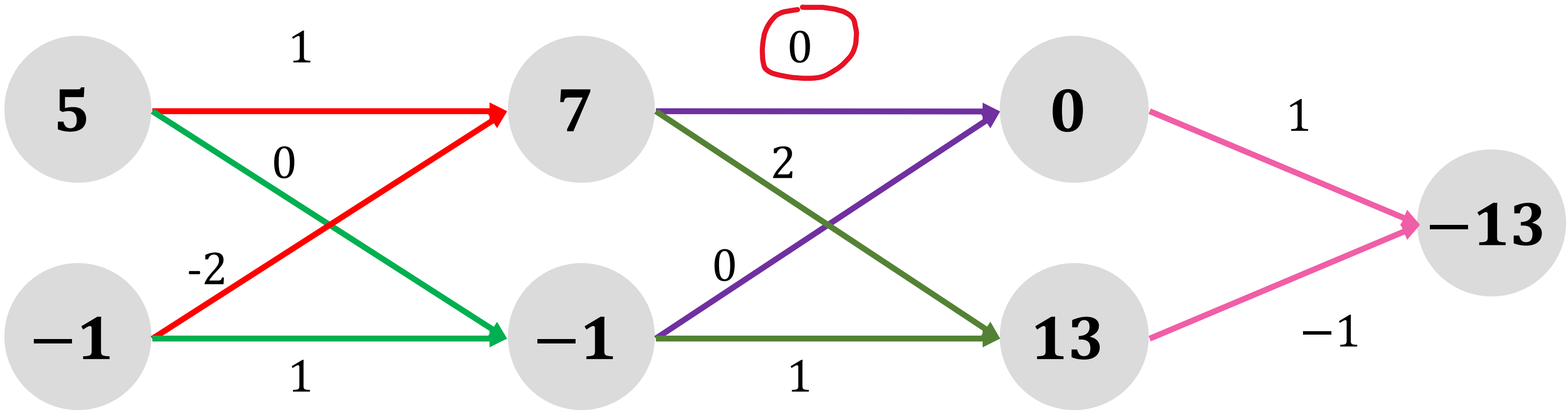
Как считать производные?



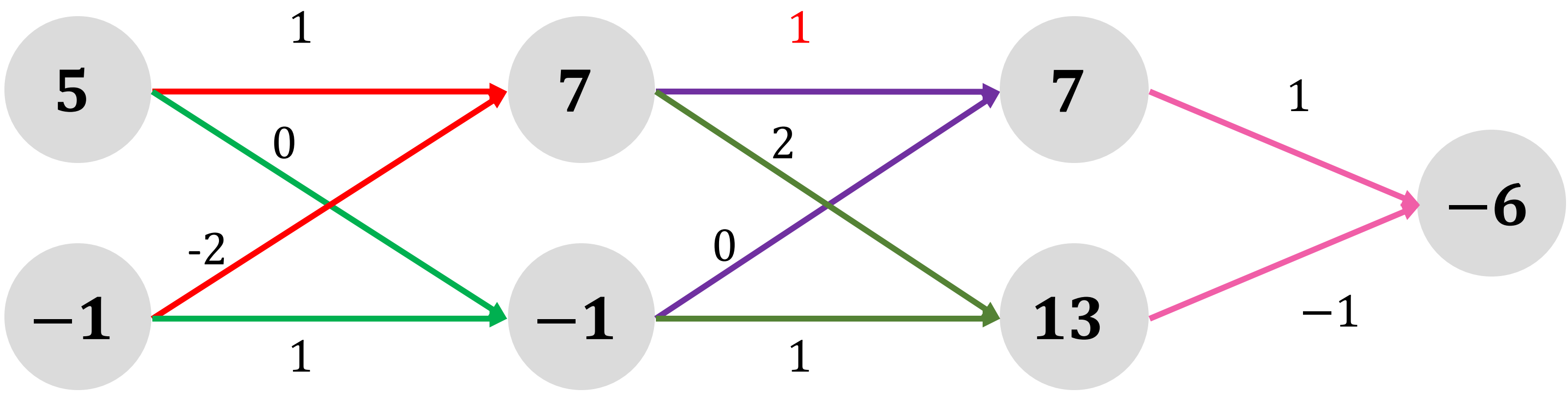
Как считать производные?



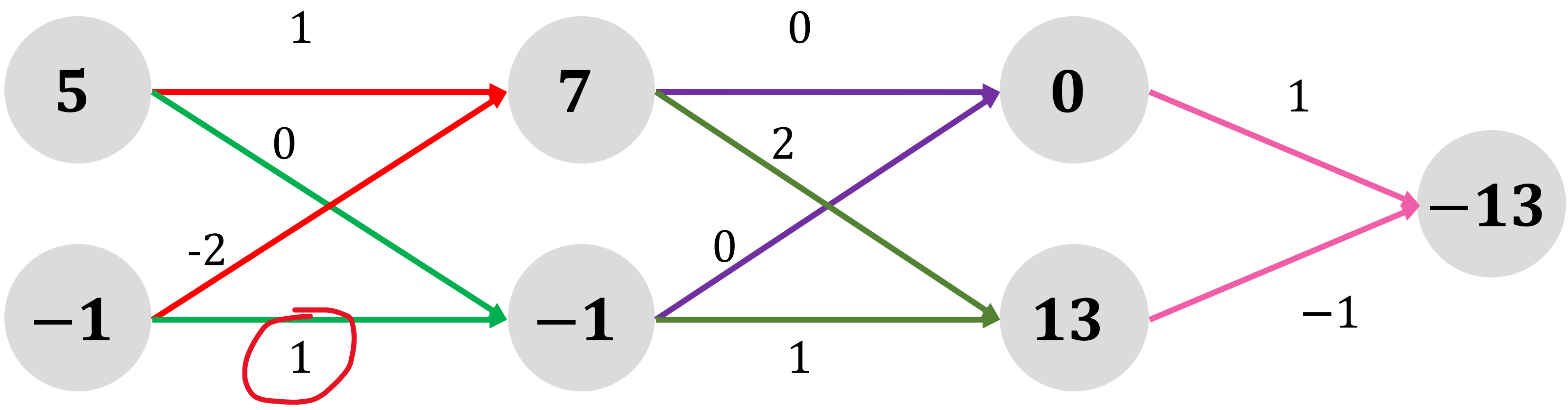
Как считать производные?



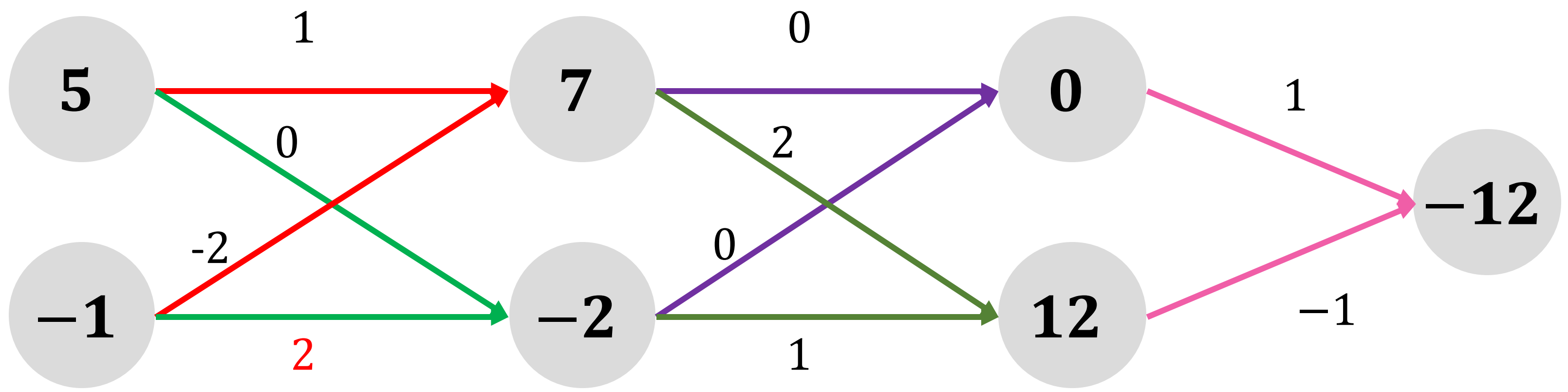
Как считать производные?



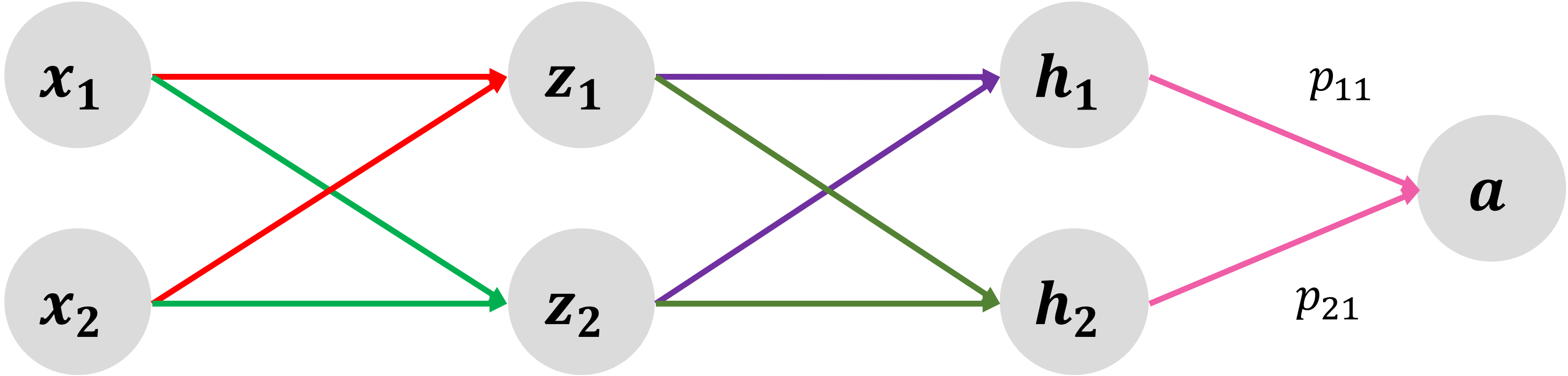
Как считать производные?



Как считать производные?



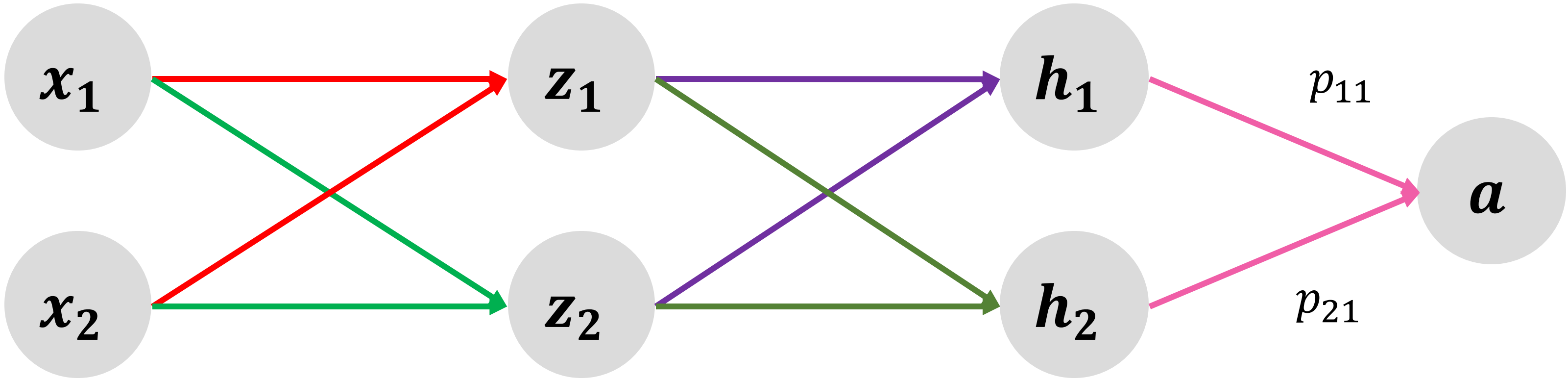
Как считать производные?



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

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

Как считать производные?

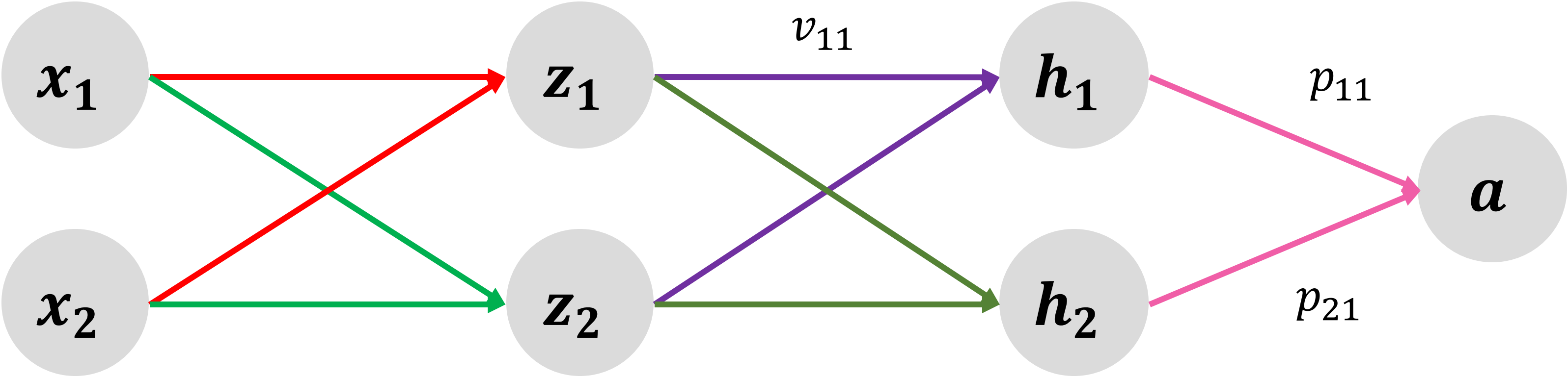


$$a(x) = p_{11}h_1(x) + p_{21}h_2(x)$$

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

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

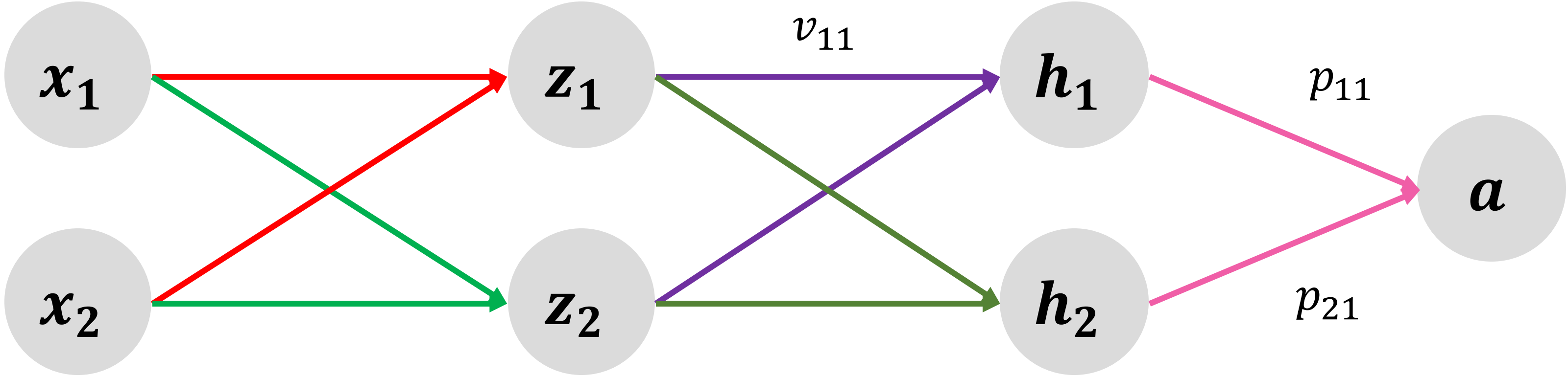
Как считать производные?



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

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

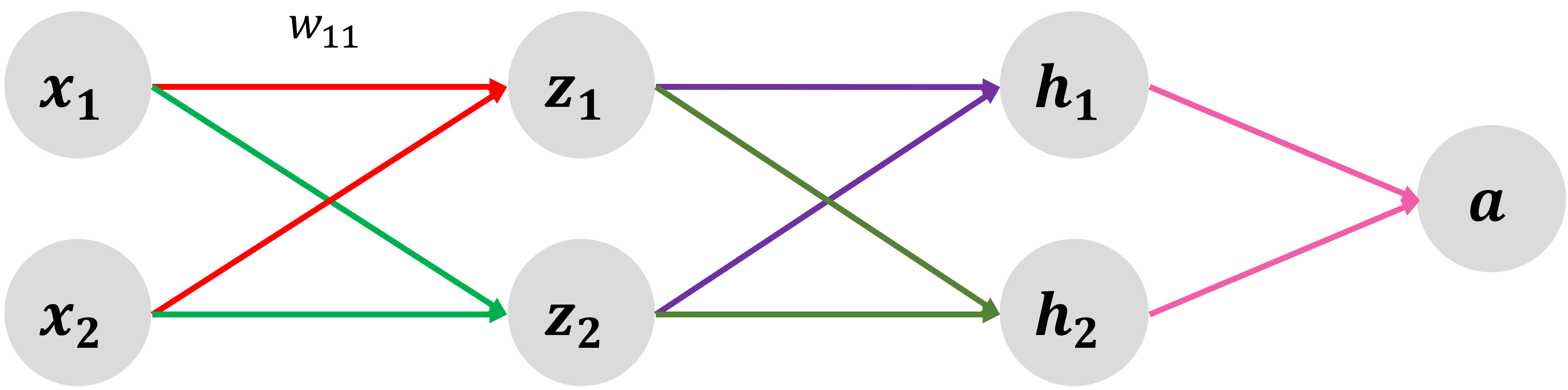
Как считать производные?



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

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

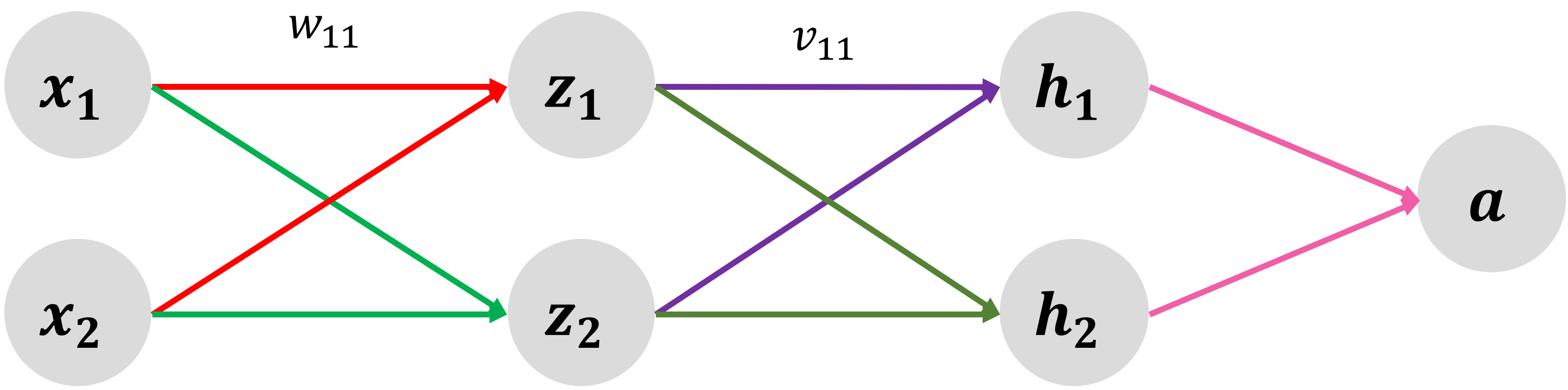
Как считать производные?



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

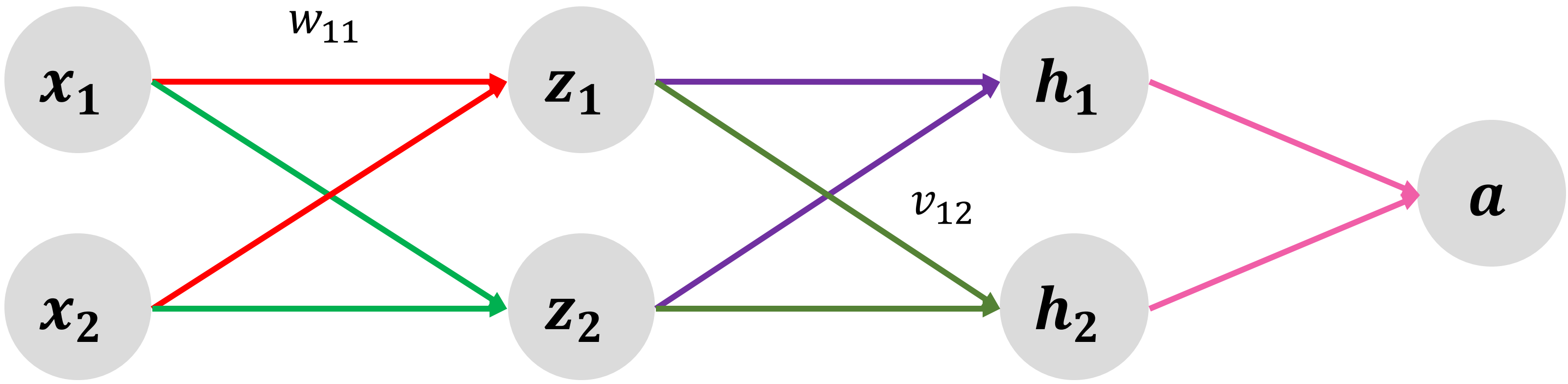
- Показывает, как сильно изменится a при изменении w_{11}

Как считать производные?



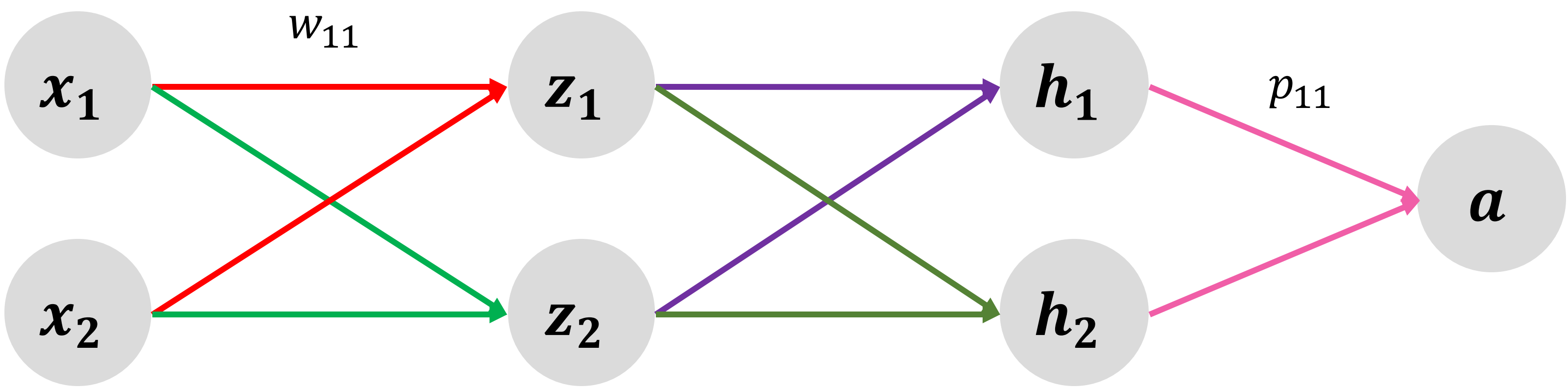
- Как сильно изменится a при изменении w_{11}?
- Влияет ли на это v_{11}?

Как считать производные?



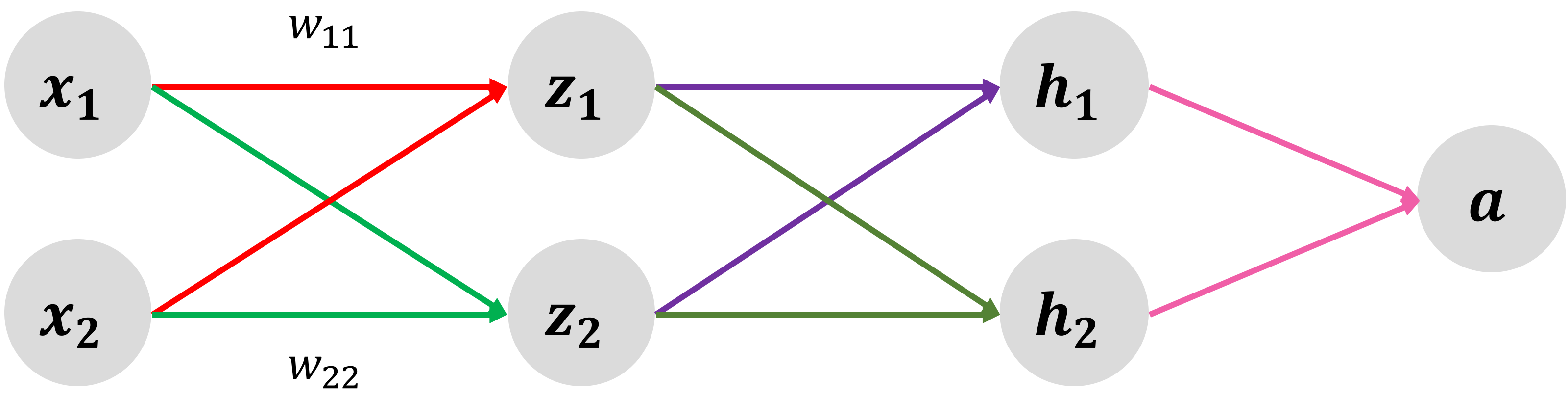
- Как сильно изменится a при изменении w_{11}?
- Влияет ли на это v_{12}?

Как считать производные?



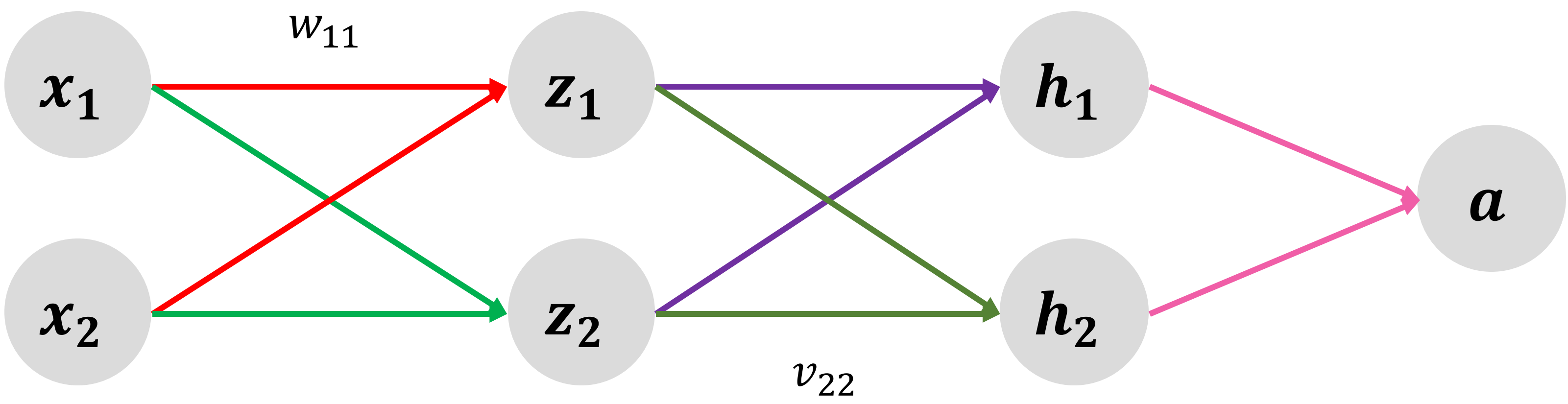
- Как сильно изменится a при изменении w_{11}?
- Влияет ли на это p_{11}?

Как считать производные?



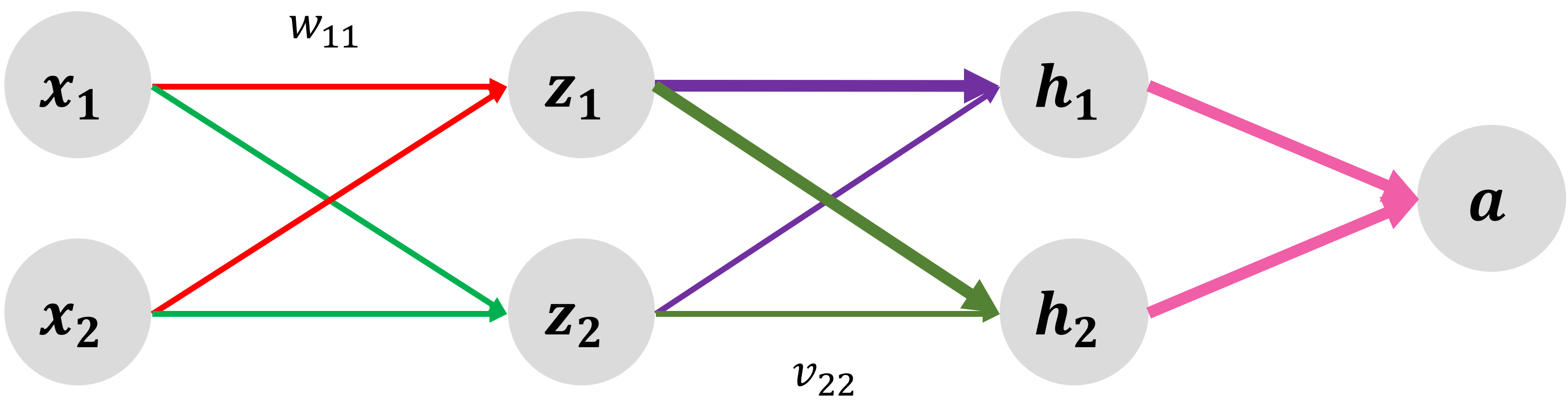
- Как сильно изменится a при изменении w_{11}?
- Влияет ли на это w_{22}?

Как считать производные?



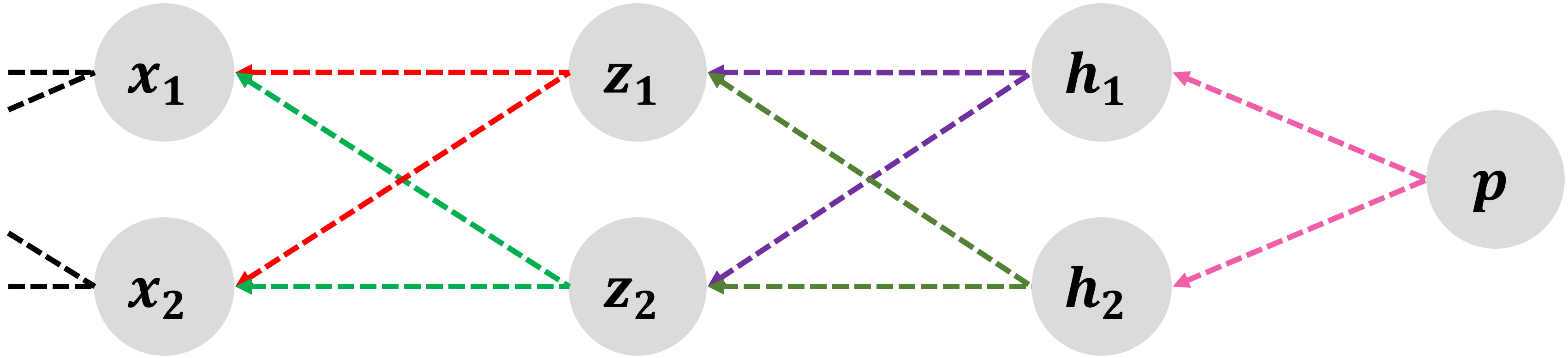
- Как сильно изменится a при изменении w_{11}?
- Влияет ли на это v_{22}?

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)

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

Резюме

- Backprop — метод вычисления градиентов для нейронных сетей
- Градиент ошибки по слою зависит от всех более поздних слоёв
- Требуется много вычислений