

# Машинное обучение

## Лекция 4 Линейная регрессия

Андрей Нарцев  
[andrei.nartsev@gmail.com](mailto:andrei.nartsev@gmail.com)  
[anartsev@hse.ru](mailto:anartsev@hse.ru)

НИУ ВШЭ, 2024

# Организационное

- На следующей лекции будет проверочная по первым пройденным темам  
(основная цель – откалибровать сложность)
- Скоро будет официально выдано первое ДЗ  
(не нужно стесняться задавать вопросы ассистентам и семинаристам)

# План лекции

## Линейная регрессия:

- Основные понятия (remind)
- Минимизация среднеквадратичной ошибки
- Особенности применения линейной регрессии (remind)
- Градиентный спуск

# ЛИНЕЙНАЯ РЕГРЕССИЯ

# ЛИНЕЙНАЯ РЕГРЕССИЯ

Пример (напоминание):

Предположим, что мы хотим предсказать *стоимость дома*  $y$  по его *площади* ( $x_1$ ) и *количеству комнат* ( $x_2$ ).

Линейная модель для предсказания стоимости:

$$a(x) = w_0 + w_1x_1 + w_2x_2,$$

где  $w_0, w_1, w_2$  -

параметры модели (*веса*).

# ЛИНЕЙНАЯ РЕГРЕССИЯ

Пример (напоминание):

Предположим, что мы хотим предсказать *стоимость дома*  $y$  по его *площади* ( $x_1$ ) и *количеству комнат* ( $x_2$ ).

Линейная модель для предсказания стоимости:

$$a(x) = w_0 + w_1x_1 + w_2x_2,$$

где  $w_0, w_1, w_2$  -

параметры модели (*веса*).



Общий вид (линейная регрессия):

$$a(x) = w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n,$$

где  $x_1, \dots, x_n$  - признаки объекта  $x$ .

# ЛИНЕЙНАЯ РЕГРЕССИЯ

Линейная регрессия:

$$a(x) = w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n$$

# ЛИНЕЙНАЯ РЕГРЕССИЯ

Линейная регрессия:

$$a(x) = w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n$$

- сокращенная запись:

$$a(x) = w_0 + \sum_{j=1}^n w_jx_j$$



# ЛИНЕЙНАЯ РЕГРЕССИЯ

Линейная регрессия:

$$a(x) = w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n$$

- сокращенная запись:

$$a(x) = w_0 + \sum_{j=1}^n w_j x_j$$

- запись через скалярное произведение (с добавлением признака  $x_0 = 1$ ):

$$a(x) = w_0 \cdot 1 + \sum_{j=1}^n w_j x_j = \sum_{j=0}^n w_j x_j = (w, x)$$

# ЛИНЕЙНАЯ РЕГРЕССИЯ

Линейная регрессия:

$$a(x) = w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n$$

- сокращенная запись:

$$a(x) = w_0 + \sum_{j=1}^n w_j x_j$$

- запись через скалярное произведение (с добавлением признака  $x_0 = 1$ ):

$$a(x) = w_0 \cdot 1 + \sum_{j=1}^n w_j x_j = \sum_{j=0}^n w_j x_j = (w, x) \leftrightarrow a(x) = (w, x)$$

# ЛИНЕЙНАЯ РЕГРЕССИЯ

Линейная регрессия:

$$a(x) = w_0 + \sum_{j=1}^n w_j x_j = (w, x)$$

Обучение линейной регрессии - минимизация  
среднеквадратичной ошибки:

$$Q(a, X) = \frac{1}{l} \sum_{i=1}^l (a(x_i) - y_i)^2 = \frac{1}{l} \sum_{i=1}^l ((w, x_i) - y_i)^2 \rightarrow \min_w$$

(здесь  $l$  – количество объектов)

ПОЧЕМУ MSE?

# ВЕРОЯТНОСТНАЯ ПОСТАНОВКА

- Даже если целевая переменная линейно зависит от признаков, то идеальной модели (с вероятностью 1) не существует, то есть реальные ответы будут (несильно) отличаться от предсказаний, поэтому мы пишем

$$y \approx (w, x)$$

- Второй подход заключается в том, что мы объясняем неидеальность прогнозом неполнотой данных, или же шумами в данных. Тогда формула переписывается со знаком “=”:

$$y = (w, x) + \varepsilon,$$

где  $\varepsilon$  — шум в данных.

# ВЕРОЯТНОСТНАЯ ПОСТАНОВКА

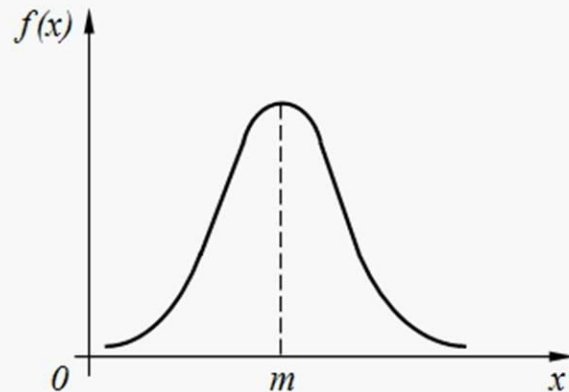
$$y = (w, x) + \varepsilon$$

- Шум в данных обычно имеет некоторое распределение. В большинстве реальных задач считается, что

$$\varepsilon \sim N(0, \sigma^2).$$

- Отсюда получаем, что  
 $y \sim N((w, x), \sigma^2).$

*График плотности нормального  
распределения*



# ВЕРОЯТНОСТНАЯ ПОСТАНОВКА

$$y \sim N((w, x), \sigma^2)$$

Это означает, что вероятность наблюдать  $y$  при данных значениях  $x$  равна

$$p(y|x, w) \sim N((w, x), \sigma^2)$$

**Мы хотим подобрать оптимальные веса. Что это такое?**

Мы хотим подобрать такой вектор  $w$ , что вероятность наблюдать некоторое значение  $y$  при наблюдаемых  $x$  максимальна.

# МЕТОД МАКСИМУМА ПРАВДОПОДОБИЯ

**Мы хотим подобрать оптимальные веса. Что это такое?**

Мы хотим подобрать такой вектор  $w$ , что вероятность наблюдать некоторое значение  $y$  при наблюдаемых  $x$  максимальна.

Запишем это желание сразу для всех объектов выборки (в предположении, что объекты независимы):

$$p(\mathbf{y}|\mathbf{X}, w) = p(y_1|x_1, w) \cdot p(y_2|x_2, w) \cdot \dots \cdot p(y_i|x_i, w) \cdot \dots \rightarrow \max_w$$

Величина  $p(\mathbf{y}|\mathbf{X}, w)$  называется ***функцией правдоподобия (или правдоподобием) выборки***.



# ММП ДЛЯ ЛИНЕЙНОЙ РЕГРЕССИИ

Модель данных с некоррелированным гауссовским шумом:

$$y_i = (w, x_i) + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2), i = 1, \dots, l$$

# ММП ДЛЯ ЛИНЕЙНОЙ РЕГРЕССИИ

Модель данных с некоррелированным гауссовским шумом:

$$y_i = (w, x_i) + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2), i = 1, \dots, l$$

$$\text{Тогда } y_i \sim N((w, x_i), \sigma^2), i = 1, \dots, l$$

# ММП ДЛЯ ЛИНЕЙНОЙ РЕГРЕССИИ

Модель данных с некоррелированным гауссовским шумом:

$$y_i = (w, x_i) + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2), i = 1, \dots, l$$

$$\text{Тогда } y_i \sim N((w, x_i), \sigma^2), i = 1, \dots, l$$

Метод максимума правдоподобия (ММП):

$$L(y_1, \dots, y_l | w) = \prod_{i=1}^l \frac{1}{\sigma \sqrt{2\pi}} \exp \left( -\frac{1}{2\sigma^2} (y_i - (w, x_i))^2 \right) \rightarrow \max_w$$

# ММП ДЛЯ ЛИНЕЙНОЙ РЕГРЕССИИ

Модель данных с некоррелированным гауссовским шумом:

$$y_i = (w, x_i) + \varepsilon_i, \varepsilon_i \sim N(0, \sigma^2), i = 1, \dots, l$$

$$\text{Тогда } y_i \sim N((w, x_i), \sigma^2), i = 1, \dots, l$$

Метод максимума правдоподобия (ММП):

$$L(y_1, \dots, y_l | w) = \prod_{i=1}^l \frac{1}{\sigma \sqrt{2\pi}} \exp \left( -\frac{1}{2\sigma^2} (y_i - (w, x_i))^2 \right) \rightarrow \max_w$$

$$-\ln L(y_1, \dots, y_l | w) = \text{const} + \frac{1}{2\sigma^2} \sum_{i=1}^l (y_i - (w, x_i))^2 \rightarrow \min_w$$

В данном случае ММП совпадает с МНК.

# ОСОБЕННОСТИ ПРИМЕНЕНИЯ ЛИНЕЙНОЙ РЕГРЕССИИ

# О ПРИМЕНИМОСТИ ЛИНЕЙНОЙ МОДЕЛИ

## Пример:

Предположим, что мы хотим предсказать *стоимость дома*  $y$  по его *площади* ( $x_1$ ) и *количеству комнат* ( $x_2$ ), *району* ( $x_3$ ) и *удаленности от МКАД* ( $x_4$ ).

$$a(x) = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4.$$



# О ПРИМЕНИМОСТИ ЛИНЕЙНОЙ МОДЕЛИ

## Пример:

Предположим, что мы хотим предсказать *стоимость дома*  $y$  по его *площади* ( $x_1$ ) и *количеству комнат* ( $x_2$ ), *району* ( $x_3$ ) и *удаленности от МКАД* ( $x_4$ ).

$$a(x) = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4.$$

Проблема №1: район ( $x_3$ ) – это не число, а название района. Например, Мамыри, Дудкино, Барвиха... Что с этим делать?



# О ПРИМЕНИМОСТИ ЛИНЕЙНОЙ МОДЕЛИ

## Пример:

Предположим, что мы хотим предсказать *стоимость дома*  $y$  по его *площади* ( $x_1$ ) и *количеству комнат* ( $x_2$ ), *району* ( $x_3$ ) и *удаленности от МКАД* ( $x_4$ ).

$$a(x) = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4.$$

Проблема №1: район ( $x_3$ ) – это не число, а название района. Например, Мамыри, Дудкино, Барвиха... Что с этим делать?

Решение – one-hot encoding (ОНЕ): создаем новые числовые столбцы, каждый из которых является индикатором района.





# ONE-HOT ENCODING



Район	Мамери	Дудкино	Барвиха
Дудкино	0	1	0
Барвиха	0	0	1
Мамери	1	0	0
...	...	...	...
Барвиха	0	0	1

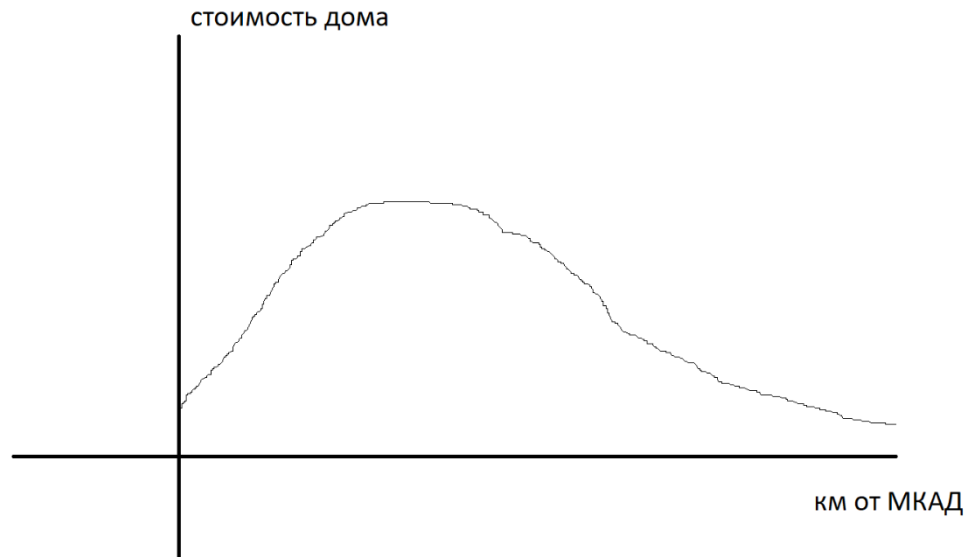
$$a(x) = w_0 + w_1 x_1 + w_2 x_2 + w_{31} x_{\text{Мамери}} + w_{32} x_{\text{Дудкино}} + w_{33} x_{\text{Барвиха}} + w_4 x_4.$$

# О ПРИМЕНИМОСТИ ЛИНЕЙНОЙ МОДЕЛИ

## Пример:

Предположим, что мы хотим предсказать *стоимость дома*  $y$  по его *площади* ( $x_1$ ) и *количеству комнат* ( $x_2$ ), *району* ( $x_3$ ) и *удаленности от МКАД* ( $x_4$ ).

Проблема №2: удаленность от МКАД ( $x_4$ ) не монотонно влияет на стоимость дома.



# О ПРИМЕНИМОСТИ ЛИНЕЙНОЙ МОДЕЛИ

Проблема №2: удаленность от МКАД ( $x_4$ ) не монотонно влияет на стоимость дома.

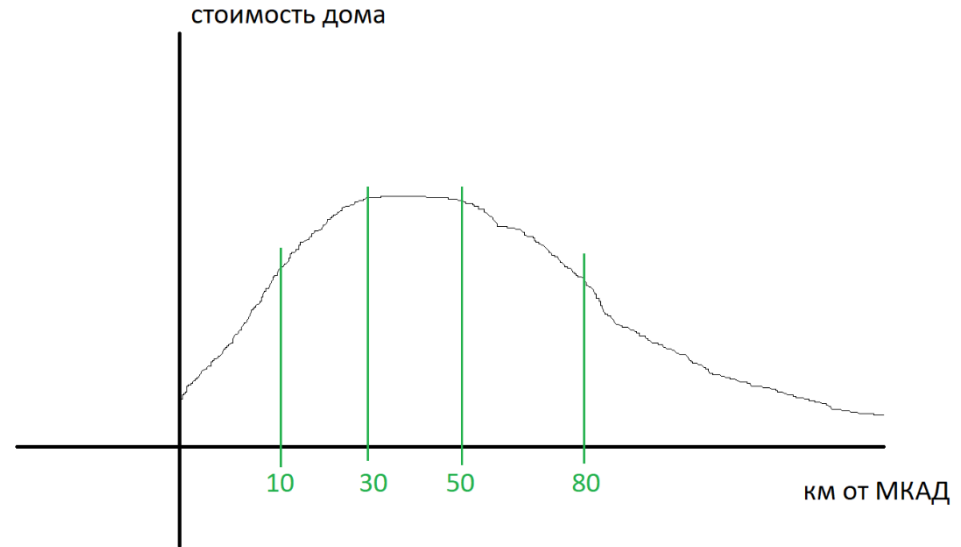
Решение – бинаризация (разбиение на бины).

Новые признаки:

- $x_{[0;10)}$  - равен 1, если дом находится в пределах 10 км от МКАД, и 0 иначе

- $x_{[10;30)}$  - равен 1, если

дом находится в пределах от 10 км до 30 км МКАД, и 0 иначе. И т.д.



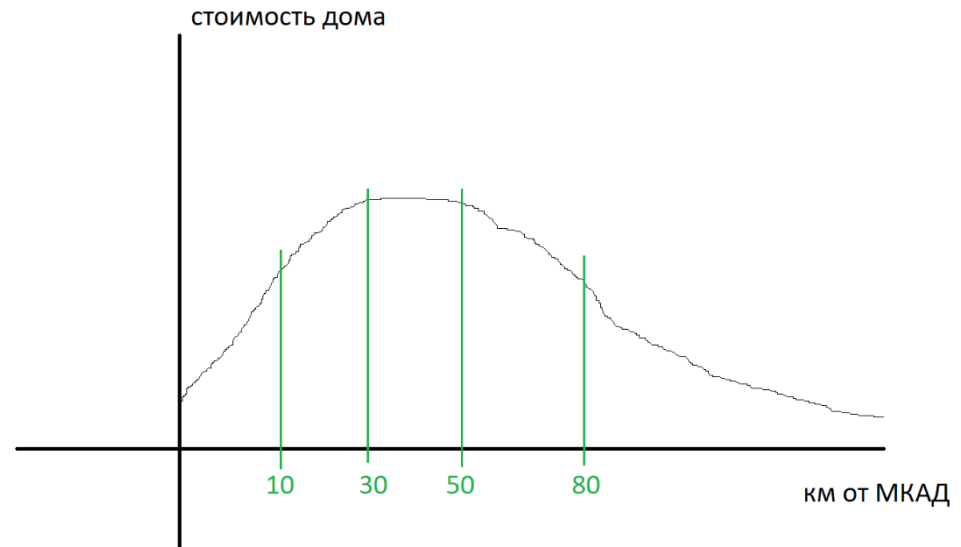
# О ПРИМЕНИМОСТИ ЛИНЕЙНОЙ МОДЕЛИ

Проблема №2: удаленность от МКАД ( $x_4$ ) не монотонно влияет на стоимость дома.

Решение – бинаризация (разбиение на бины).

Новые признаки:

- $x_{[0;10)}$  - равен 1, если дом находится в пределах 10 км от МКАД, и 0 иначе
- $x_{[10;30)}$  - равен 1, если дом находится в пределах от 10 км до 30 км МКАД, и 0 иначе. И т.д.



$$\begin{aligned} a(x) = & \\ = & w_0 + w_1x_1 + w_2x_2 + \dots + w_{41}x_{[0;10)} + w_{42}x_{[10;30)} + w_{43}x_{[30;50)} \\ & + w_{44}x_{\geq 50} \end{aligned}$$

# АНАЛИТИЧЕСКОЕ РЕШЕНИЕ ЗАДАЧИ МЕТОДА НАИМЕНЬШИХ КВАДРАТОВ (МНК)

Задача обучения линейной регрессии (в матричной форме):

$$\frac{1}{\ell} \|Xw - y\|^2 \rightarrow \min_w$$

Точное (аналитическое) решение:

$$w = (X^T X)^{-1} X^T y$$

# НЕДОСТАТКИ АНАЛИТИЧЕСКОЙ ФОРМУЛЫ

- Обращение матрицы – сложная операция ( $O(N^3)$  от числа признаков)
- Матрица  $X^T X$  может быть вырожденной или плохо обусловленной
- Если заменить среднеквадратичный функционал ошибки на другой, то скорее всего не найдем аналитическое решение

# ГРАДИЕНТНЫЙ СПУСК

# МЕТОД ГРАДИЕНТНОГО СПУСКА

- Наша задача при обучении модели – найти такие веса  $w$ , на которых достигается **минимум функции ошибки**.



# МЕТОД ГРАДИЕНТНОГО СПУСКА

- Наша задача при обучении модели – найти такие веса  $w$ , на которых достигается минимум функции ошибки.
- В простейшем случае, если ошибка среднеквадратичная, то её график – это парабола.

# МЕТОД ГРАДИЕНТНОГО СПУСКА

- Наша задача при обучении модели – найти такие веса  $w$ , на которых достигается минимум функции ошибки.
- В простейшем случае, если ошибка среднеквадратичная, то её график – это парабола.
- Идея метода градиентного спуска:

На каждом шаге (на каждой итерации метода) движемся в сторону антиградиента функции потерь!

То есть на каждом шаге движемся в направлении уменьшения ошибки.

# МЕТОД ГРАДИЕНТНОГО СПУСКА

- Наша задача при обучении модели – найти такие веса  $w$ , на которых достигается минимум функции ошибки.
- В простейшем случае, если ошибка среднеквадратичная, то её график – это парабола.
- **Идея метода градиентного спуска:**

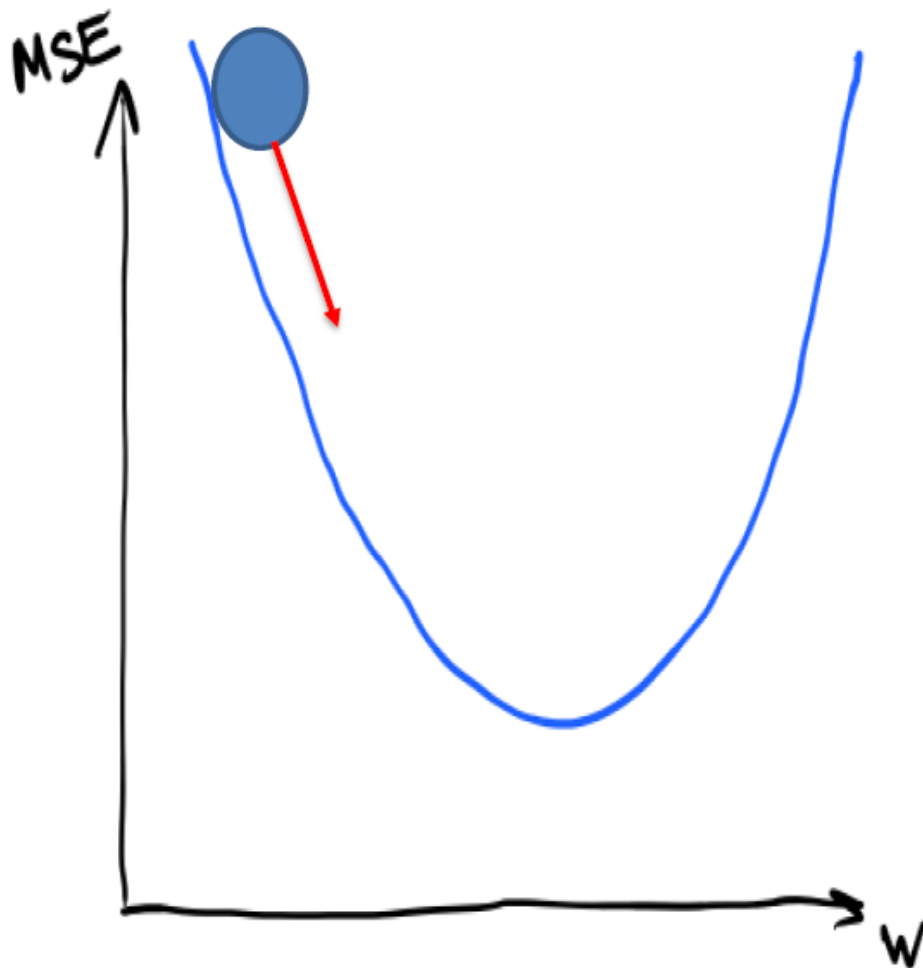
На каждом шаге (на каждой итерации метода) движемся в сторону антиградиента функции потерь!

То есть на каждом шаге движемся в направлении уменьшения ошибки.

Вектор градиента функции потерь обозначают ***grad Q*** или  ***$\nabla Q$*** .

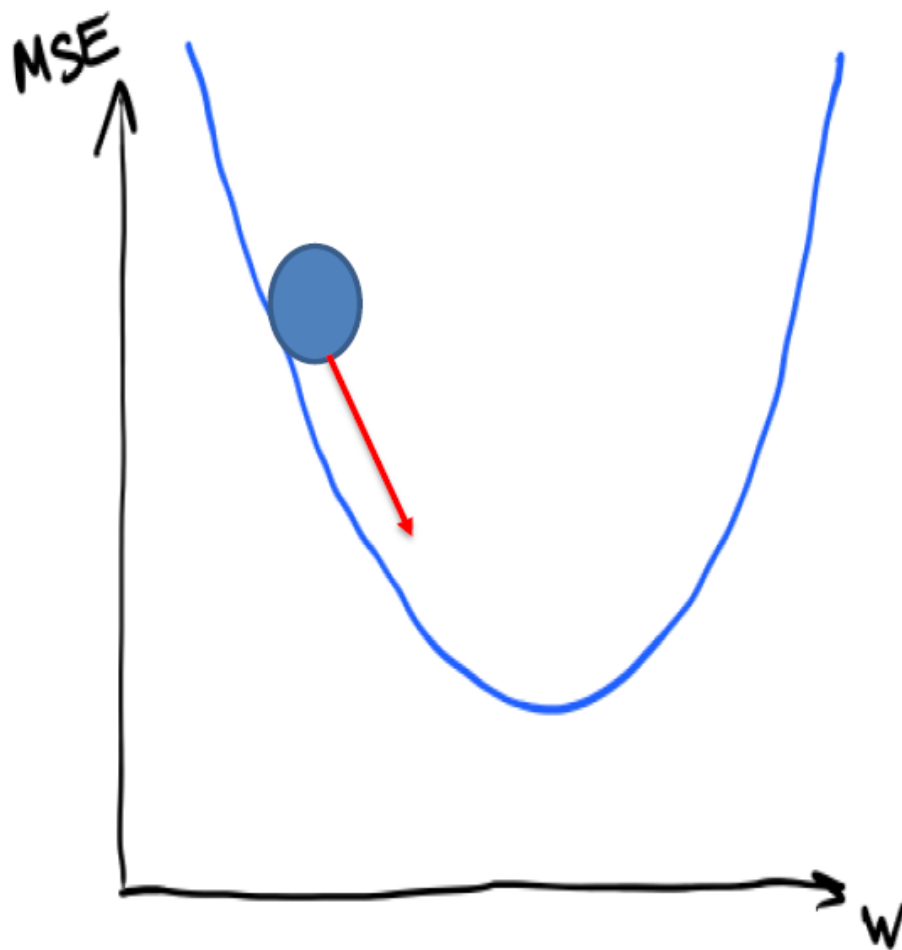
# МЕТОД ГРАДИЕНТНОГО СПУСКА

На каждом шаге (на каждой итерации метода) движемся в сторону антиградиента функции потерь!



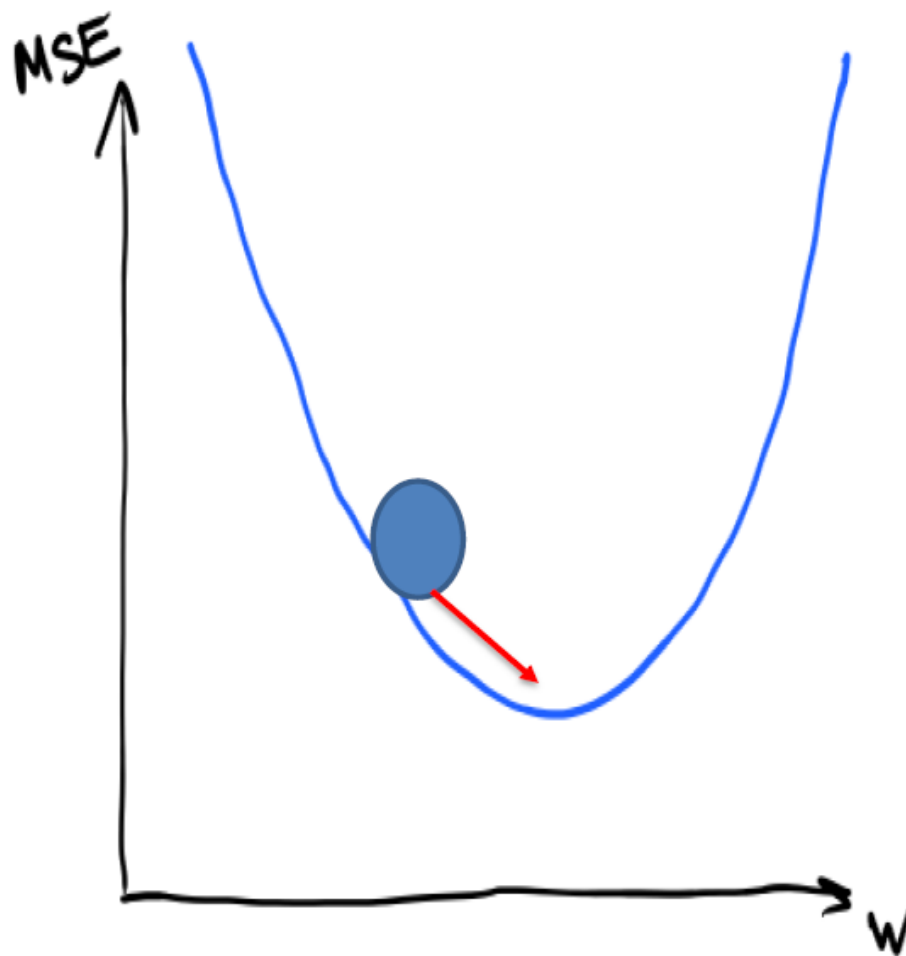
# МЕТОД ГРАДИЕНТНОГО СПУСКА

На каждом шаге (на каждой итерации метода) движемся в сторону антиградиента функции потерь!



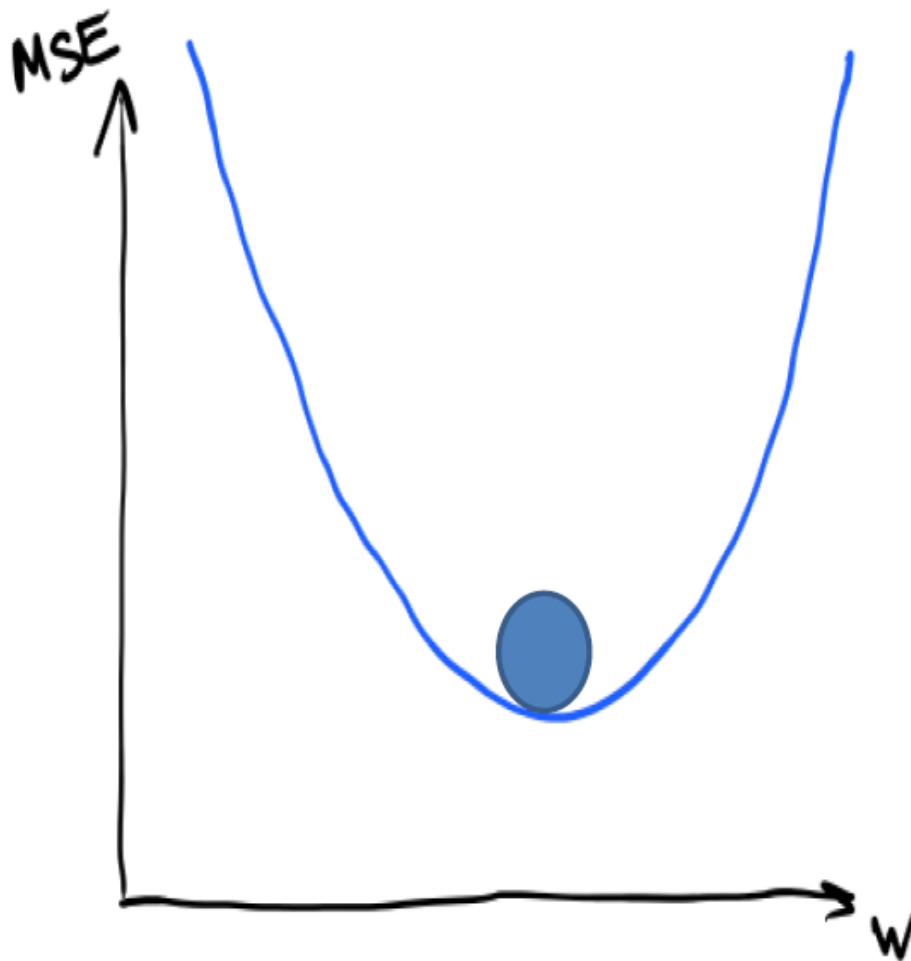
# МЕТОД ГРАДИЕНТНОГО СПУСКА

На каждом шаге (на каждой итерации метода) движемся в сторону антиградиента функции потерь!



# МЕТОД ГРАДИЕНТНОГО СПУСКА

На каждом шаге (на каждой итерации метода) движемся в сторону антиградиента функции потерь!



# МЕТОД ГРАДИЕНТНОГО СПУСКА

**Метод градиентного спуска (одномерный случай):**

Пусть у нас только один вес -  $w$ .

Тогда при добавлении к весу  $w$  слагаемого  $-\frac{\partial Q}{\partial w}$  функция  $Q(w)$  убывает.



# МЕТОД ГРАДИЕНТНОГО СПУСКА

## Метод градиентного спуска (одномерный случай):

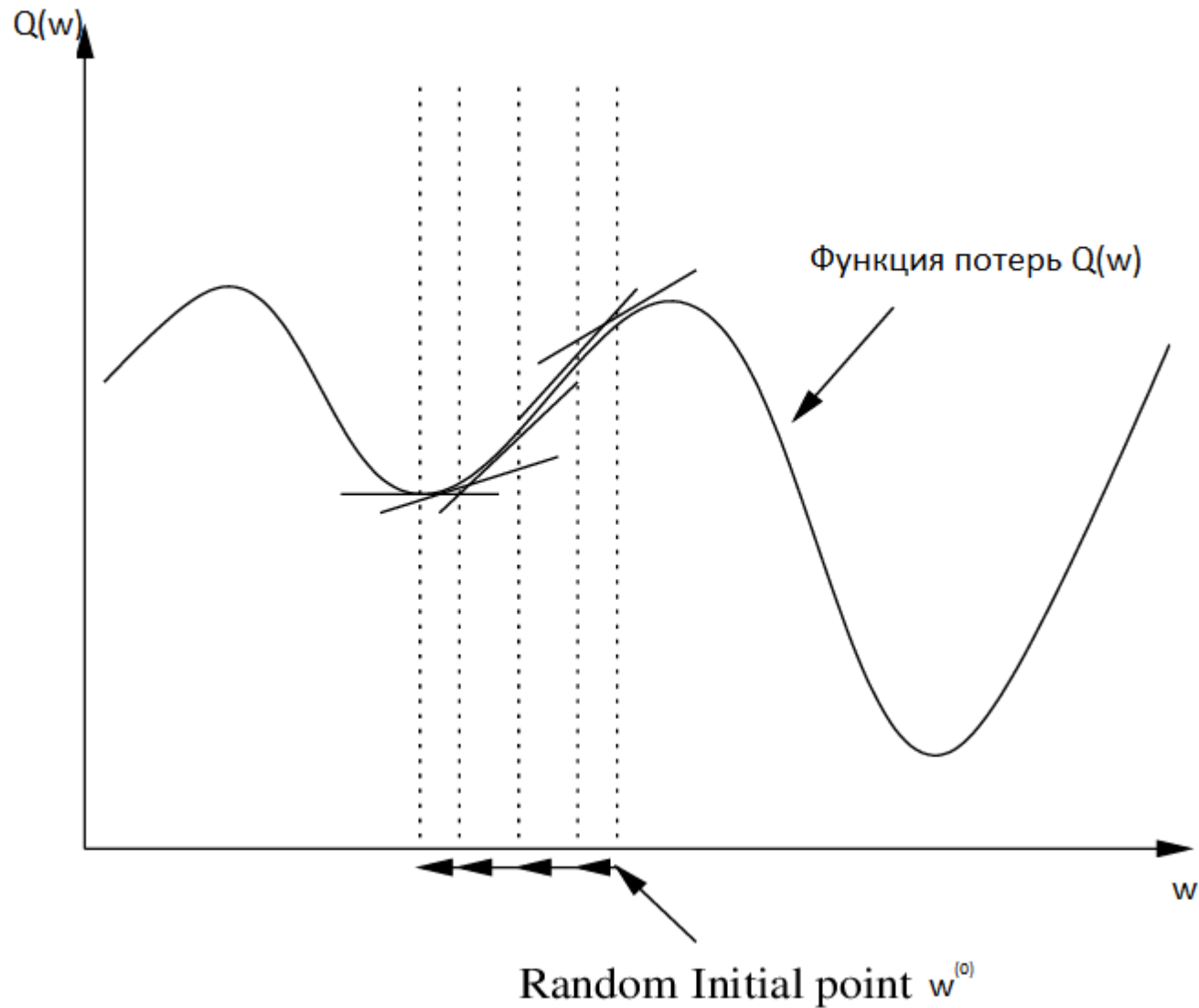
Пусть у нас только один вес -  $w$ .

Тогда при добавлении к весу  $w$  слагаемого  $-\frac{\partial Q}{\partial w}$  функция  $Q(w)$  убывает.

- Инициализируем вес  $w^{(0)}$ .
- На каждом следующем шаге обновляем вес, добавляя  $-\frac{\partial Q}{\partial w}(w^{(k-1)})$ :

$$w^{(k)} = w^{(k-1)} - \frac{\partial Q}{\partial w}(w^{(k-1)})$$

# МЕТОД ГРАДИЕНТНОГО СПУСКА



# МЕТОД ГРАДИЕНТНОГО СПУСКА

**Метод градиентного спуска (общий случай случай):**

Пусть  $w_0, w_1, \dots, w_n$  - веса, которые мы ищем.

Тогда  $\nabla Q(w) = \left\{ \frac{\partial Q}{\partial w_0}, \frac{\partial Q}{\partial w_1}, \dots, \frac{\partial Q}{\partial w_n} \right\}$

# МЕТОД ГРАДИЕНТНОГО СПУСКА

Формулу для обновления весов можно записать в векторном виде:

- Инициализируем веса  $w^{(0)}$ .
- На каждом следующем шаге обновляем веса по формуле:

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

# МЕТОД ГРАДИЕНТНОГО СПУСКА

Формулу для обновления весов можно записать в векторном виде:

- Инициализируем веса  $\mathbf{w}^{(0)}$ .
- На каждом следующем шаге обновляем веса по формуле:

$$\mathbf{w}^{(k)} = \mathbf{w}^{(k-1)} - \nabla Q(\mathbf{w}^{(k-1)})$$

В формулу обычно добавляют параметр  $\eta$  – величина градиентного шага (learning rate). Он отвечает за скорость движения в сторону антиградиента:

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

# МЕТОД ГРАДИЕНТНОГО СПУСКА

Формулу для обновления весов можно записать в векторном виде:

- Инициализируем веса  $\mathbf{w}^{(0)}$ .
- На каждом следующем шаге обновляем веса по формуле:

$$\mathbf{w}^{(k)} = \mathbf{w}^{(k-1)} - \nabla Q(\mathbf{w}^{(k-1)})$$

В формулу обычно добавляют параметр  $\eta$  – величина градиентного шага (learning rate). Он отвечает за скорость движения в сторону антиградиента:

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

Если функция  $Q(\mathbf{w})$  выпуклая и гладкая, а также имеет минимум в точке  $\mathbf{w}^*$ , то метод градиентного спуска при аккуратно подобранном  $\eta$  через некоторое число шагов гарантированно попадет в малую окрестность точки  $\mathbf{w}^*$ .

# ВАРИАНТЫ ИНИЦИАЛИЗАЦИИ ВЕСОВ

- $w_j = 0, j = 1, \dots, n$

- Небольшие случайные значения:

$$w_j := \text{random}(-\varepsilon, \varepsilon)$$

- Обучение по небольшой случайной подвыборке объектов
- Мультистарт: многократный запуск из разных случайных начальных приближений и выбор лучшего решения

# КРИТЕРИИ ОСТАНОВА

- $|Q(w^{(k)}) - Q(w^{(k-1)})| < \varepsilon$

- $\|w^{(k)} - w^{(k-1)}\| < \varepsilon$

- $\|\nabla Q(w^{(k)})\| < \varepsilon$



# ГРАДИЕНТНЫЙ ШАГ

В общем случае градиентный шаг может зависеть от номера итерации, тогда будем писать не  $\eta$ , а  $\eta_k$ .

- $\eta_k = c$
- $\eta_k = \frac{1}{k}$
- $\eta_k = \lambda \left( \frac{s_0}{s_0 + k} \right)^p$ ,  $\lambda, s_0, p$  - параметры

# ОДИН ИЗ НЕДОСТАТКОВ ГРАДИЕНТНОГО СПУСКА

(с точки зрения реализации)

- На каждом шаге для вычисления  $\nabla Q(w)$  мы вычисляем производную по каждому весу от каждого объекта. То есть вычисляем целую матрицу производных — это затратно и по времени, и по памяти.

# СТОХАСТИЧЕСКИЙ ГРАДИЕНТНЫЙ СПУСК

Stochastic gradient descent (SGD):

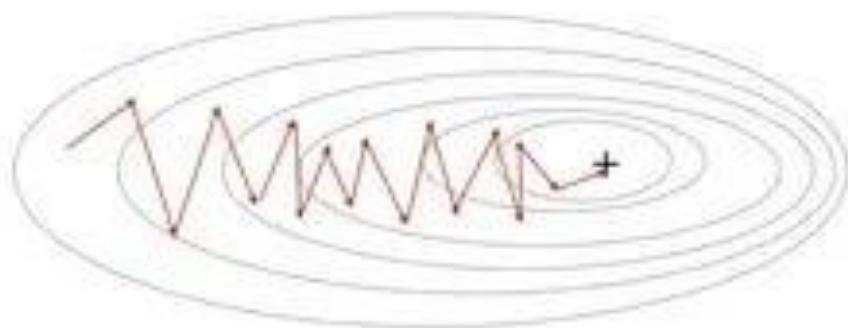
- на каждом шаге выбираем ***один случайный объект*** и сдвигаемся в сторону антиградиента по этому объекту:

$$w^{(k)} = w^{(k-1)} - \eta_k \cdot \nabla q_{i_k}(w^{(k-1)}),$$

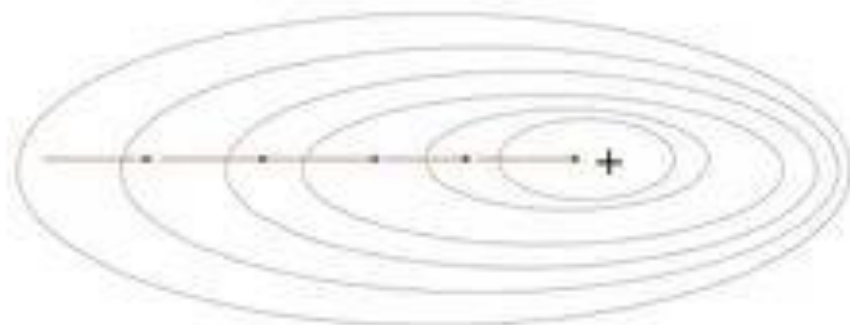
где  $\nabla q_{i_k}(w^{(k-1)})$  - градиент функции потерь, вычисленный только по объекту с номером  $i_k$  (а не по всей обучающей выборке).

# СТОХАСТИЧЕСКИЙ ГРАДИЕНТНЫЙ СПУСК

Stochastic Gradient Descent



Gradient Descent



Если функция  $Q(w)$  выпуклая и гладкая, а также имеет минимум в точке  $w^*$ , то метод стохастического градиентного спуска при аккуратно подобранном  $\eta$  через некоторое число шагов гарантированно попадет в малую окрестность точки  $w^*$ . Однако, сходится метод медленнее, чем обычный градиентный спуск

# MINI-BATCH GRADIENT DESCENT

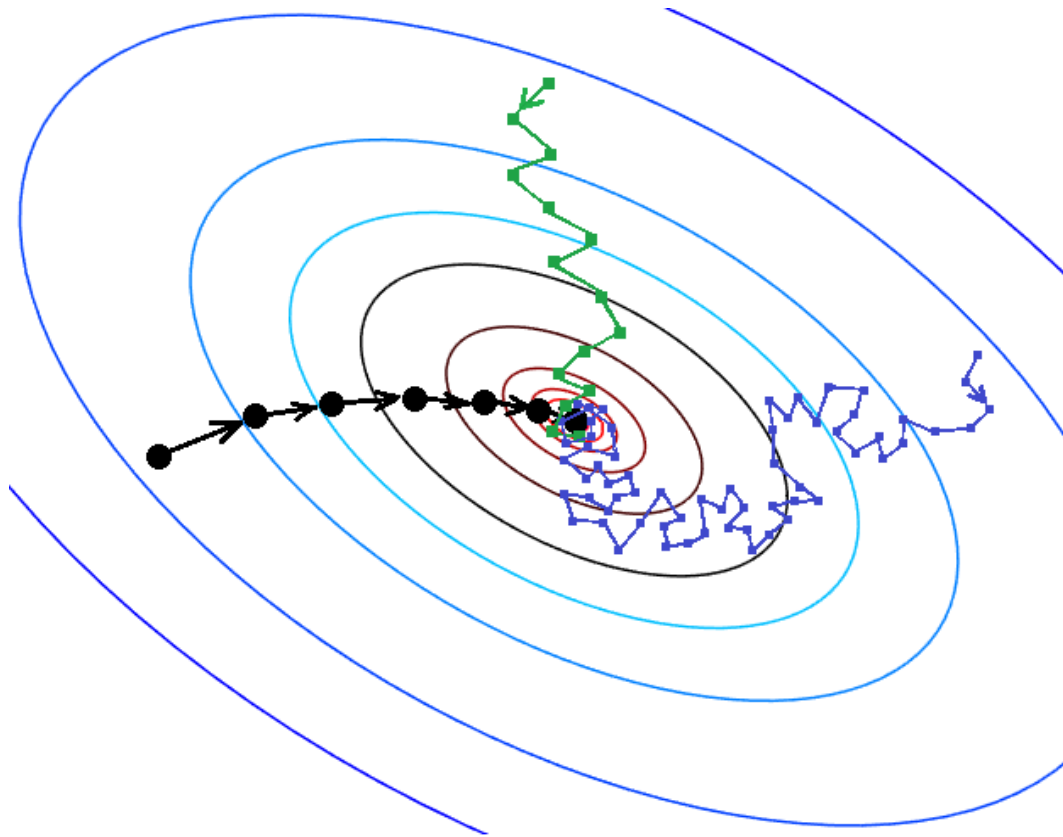
Промежуточное решение между классическим градиентным спуском и стохастическим вариантом.

- Выбираем batch size (например, 32, 64 и т.д.). Разбиваем все пары объект-ответ на группы размера batch size.
- На  $i$ -й итерации градиентного спуска вычисляем  $\nabla Q(w)$  только по объектам  $i$ -го батча:

$$w^{(k)} = w^{(k-1)} - \eta_k \cdot \nabla Q_i(w^{(k-1)}),$$

где  $\nabla Q_i(w^{(k-1)})$  - градиент функции потерь, вычисленный по объектам из  $i$ -го батча.

# ВАРИАНТЫ ГРАДИЕНТНОГО СПУСКА



## Batch GD

- Slowest
- Perfect gradient

## Stochastic GD

- Fastest
- Rough-estimate grad

## Mini-batch GD

- Compromise

# В следующих сериях

## Линейная регрессия:

- Вывод аналитического решения в матричном виде для MSE
- Модификации градиентного спуска
- Метрики в задачах регрессии
- Регуляризация линейных моделей для борьбы с переобучением

## Линейная классификация