

Занятие 2

Линейные методы

регрессии. Часть 1.

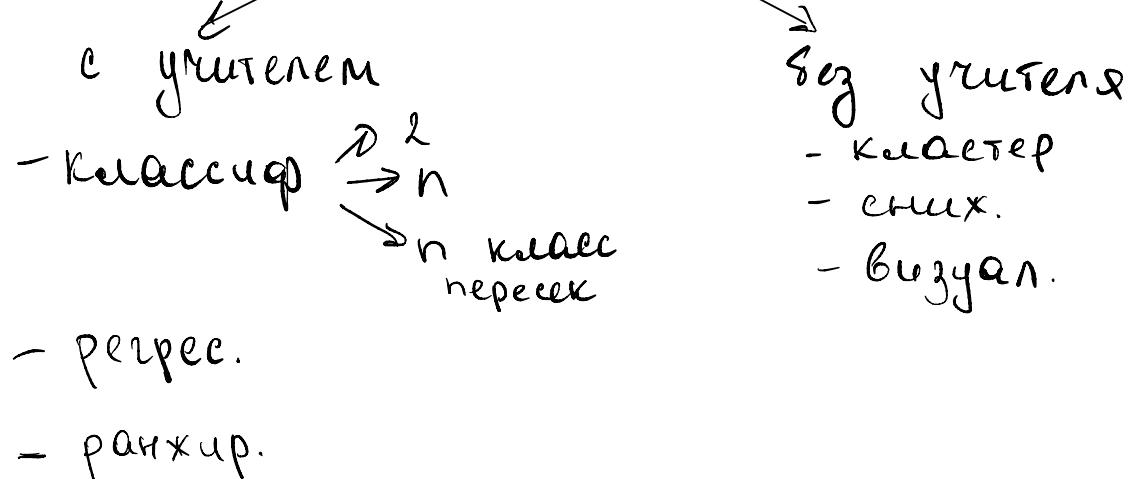
Юлия Конюшенко

тг: @ko_iulia

koniushenko.iun@phystech.edu

ВШЭ, 2023

ML



ПЛАН ЛЕКЦИИ

- Отложенная выборка и переобучение
- Линейная регрессия
- Почему MSE? Вероятностное объяснение
- Особенности применения линейной регрессии
- Градиентный спуск
- Модификации градиентного спуска (если успеем)

ОТЛОЖЕННАЯ ВЫБОРКА И ПЕРЕОБУЧЕНИЕ

ОЦЕНКА ПРЕДСКАЗАТЕЛЬНОЙ СПОСОБНОСТИ АЛГОРИТМА

- Пусть мы решаем задачу *предсказания стоимости дома по его признакам.*



- В обучающей выборке 1000 домов.
- Мы обучаем алгоритм по имеющимся 1000 домам. *На каких объектах будем проверять качество алгоритма?*

ОЦЕНКА ПРЕДСКАЗАТЕЛЬНОЙ СПОСОБНОСТИ АЛГОРИТМА

- Перед началом обучения отложим часть обучающих объектов и не будем использовать их для построения модели (отложенная выборка).

какая пропорция ?

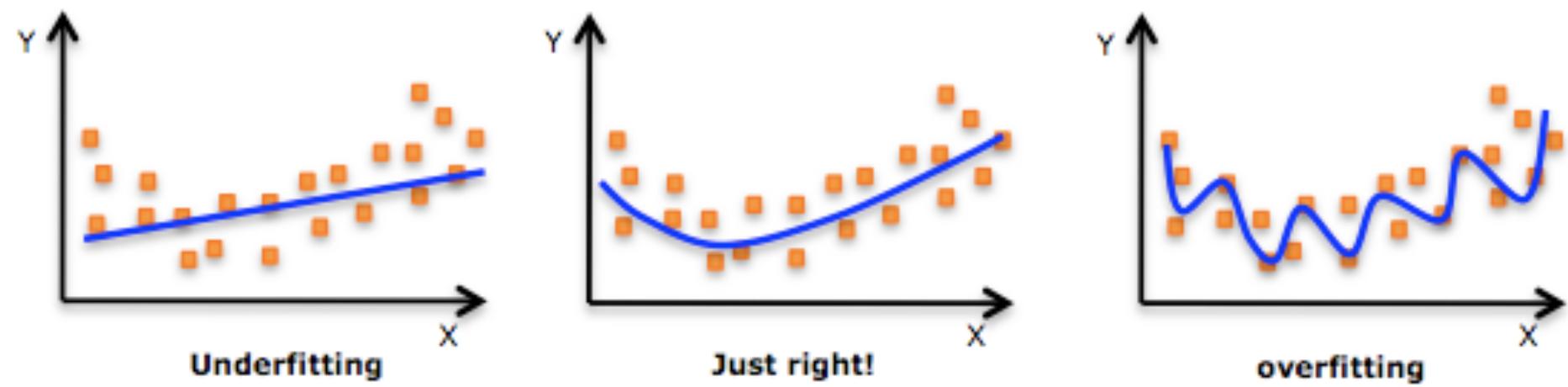
- 80 / 20
- 70 / 30
- 50 / 50
- 20 / 80
- 99 / 1
- 1 / 99



ОТЛОЖЕННАЯ ВЫБОРКА

- Перед началом обучения отложим часть обучающих объектов и не будем использовать их для построения модели (отложенная выборка).
- Тогда можно измерить качество построенной модели на отложенной выборке и оценить ее предсказательную силу.

ПЕРЕОБУЧЕНИЕ И НЕДООБУЧЕНИЕ



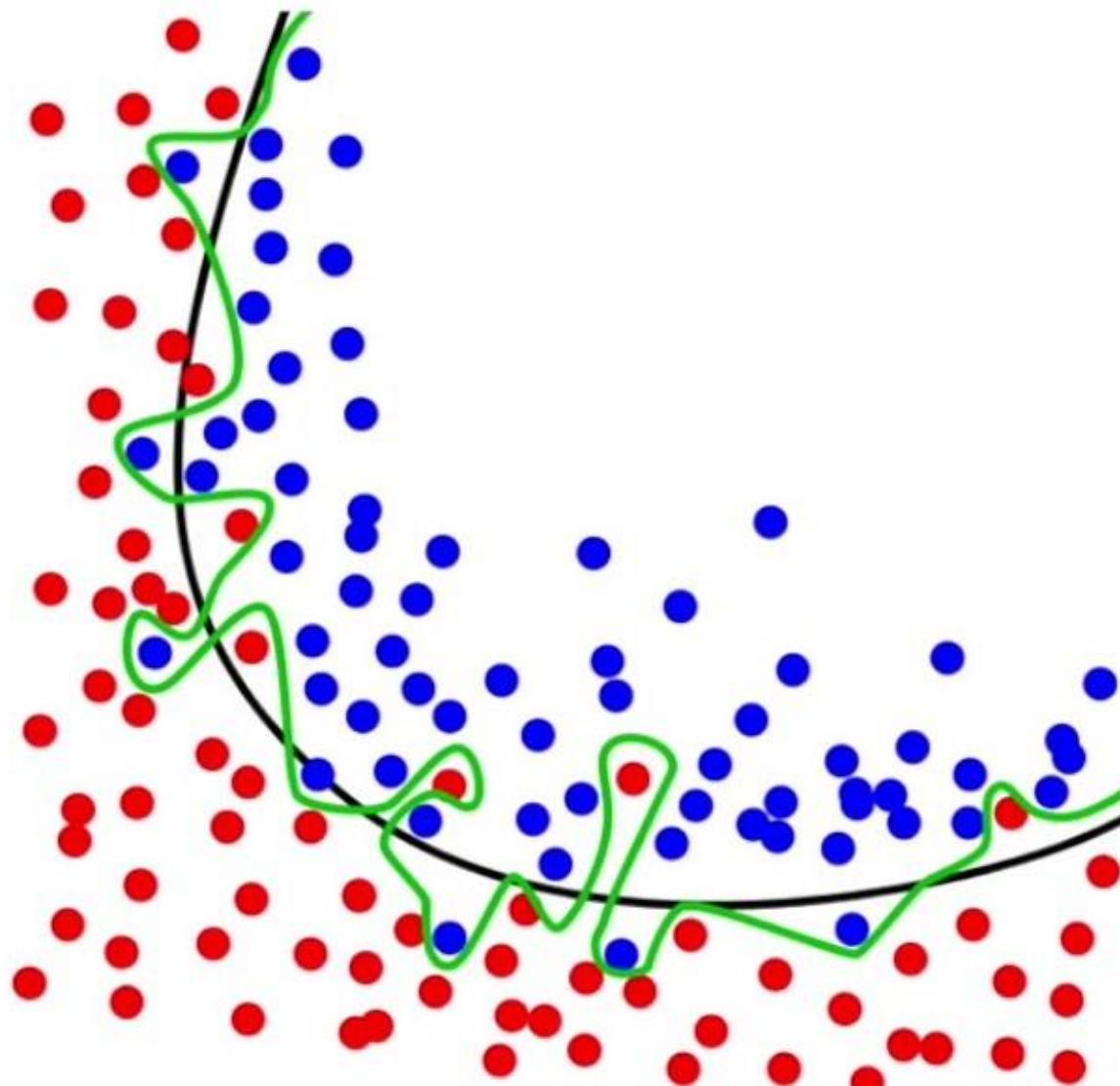
Какая проблема выражается выше?

$$y = w_0 + w_1 x_1 + w_2 x_2$$

ИЗ-ЗА ЧЕГО ВОЗНИКАЕТ ПЕРЕОБУЧЕНИЕ

- Избыточная сложность модели (большое количество весов). В этом случае лишние степени свободы в модели “тратятся” на чрезмерно точную подгонку под обучающую выборку.
- Переобучение есть всегда, когда есть оптимизация параметров по конечной (заведомо неполной) выборке.

ПРИМЕР ПЕРЕОБУЧЕНИЯ В ЗАДАЧЕ КЛАССИФИКАЦИИ



ПРИЗНАК ПЕРЕОБУЧЕНИЯ

- Если качество на отложенной выборке сильно ниже качества на обучающих данных, то происходит переобучение

1) выбираем метрику
2) считаем метрику на train
3) считаем метрику на test
4) m_{tr} vs m_{test}
 m_{tr} лучше m_{test} $\xrightarrow{\text{сильно}}$ переобучение

Метрики

Классификатор

- accuracy
- количество правильных отгадок
- recall
- precision
- f₁

регрессия

- MSE
- R²
- ...

$$\frac{\sum_i (y_{\text{pred}} - y_{\text{true}})^2}{n}$$

ЛИНЕЙНАЯ РЕГРЕССИЯ

ЛИНЕЙНАЯ РЕГРЕССИЯ

Пример (напоминание):

Предположим, что мы хотим предсказать *стоимость дома* у по его *площади (x_1)* и *количество комнат (x_2)*.

Линейная модель для предсказания стоимости:

$$a(x) = \underbrace{w_0}_{} + \underbrace{w_1}_{-}x_1 + \underbrace{w_2}_{-}x_2,$$

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 *количество комнат* (x_2).

Линейная модель для предсказания стоимости:

$$a(x) = w_0 + w_1 x_1 + w_2 x_2,$$

где w_0, w_1, w_2 -

параметры модели (веса).



Общий вид (линейная регрессия):

$$a(x) = w_0 + w_1 x_1 + w_2 x_2 + \dots + w_n x_n,$$

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:

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

Можно ли оставить в записи только \sum ?

ЛИНЕЙНАЯ РЕГРЕССИЯ

Линейная регрессия:

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

- сокращенная запись:

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

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

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

ЛИНЕЙНАЯ РЕГРЕССИЯ

Линейная регрессия:

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

- сокращенная запись:

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

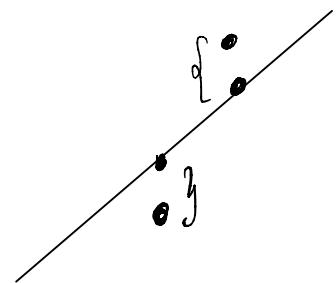
- запись через скалярное произведение (с добавлением признака $x_0 = 1$):

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

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

$$y = a(x) = (w, x)$$

- 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 неполной информацией, или же шумами в данных. Тогда формула переписывается со знаком “=”:

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

где ε – шум в данных.

ВЕРОЯТНОСТНАЯ ПОСТАНОВКА

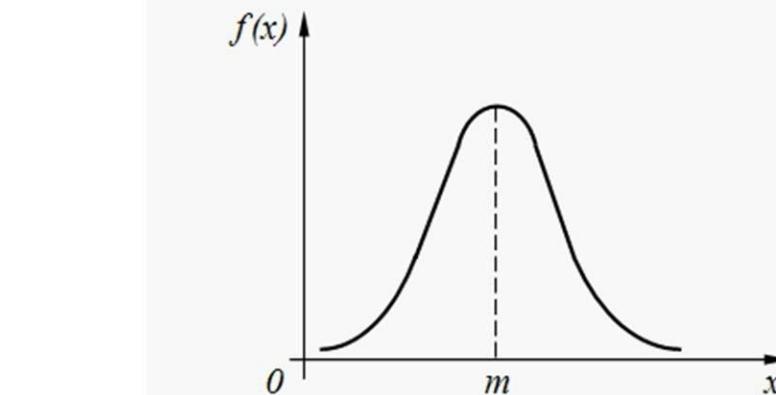
$$y = (w, x) + \varepsilon$$

- 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):

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

Величина $p(y|X, w)$ 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:

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

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

ММП ДЛЯ ЛИНЕЙНОЙ РЕГРЕССИИ

Модель данных с некоррелированным гауссовским шумом:

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

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:

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

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

Метод максимума правдоподобия (ММП):

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

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

В данном случае ММП совпадает с МНК.

$P(y | X, w)$ *хочим* $w - ?$
стабиль $P(y | X, w) \rightarrow \max \Rightarrow$ преобр + $\ln \rightarrow MSE$

ОСОБЕННОСТИ ПРИМЕНЕНИЯ ЛИНЕЙНОЙ РЕГРЕССИИ

О ПРИМЕНИМОСТИ ЛИНЕЙНОЙ МОДЕЛИ

Пример:

Предположим, что мы хотим предсказать *стоимость дома* у по его *площади (x_1)* и *количество 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* у по его *площади (x_1)* и *количество 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?

- 1) Присв. портфл. номер
- 2) OHE (one-hot encoding)



О ПРИМЕНИМОСТИ ЛИНЕЙНОЙ МОДЕЛИ

Пример:

Предположим, что мы хотим предсказать *стоимость дома* у по его *площади (x_1)* и *количество 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 (OHE)*: создаем новые числовые столбцы, каждый из которых является индикатором района.

ONE-HOT ENCODING

нужно есть $n-1$
признак



Район
Дудкино
Барвиха
Мамыри
...
Барвиха



	Мамыри	Дудкино	Барвиха
0	1	0	0
0	0	1	0
1	0	0	0
...
0	0	0	1

$$a(x) =$$

$$= w_0 + w_1 x_1 + w_2 x_2 + w_{31} x_{\text{Мамыри}} + w_{32} x_{\text{Дудкино}} + w_{33} x_{\text{Барвиха}} + w_4 x_4.$$

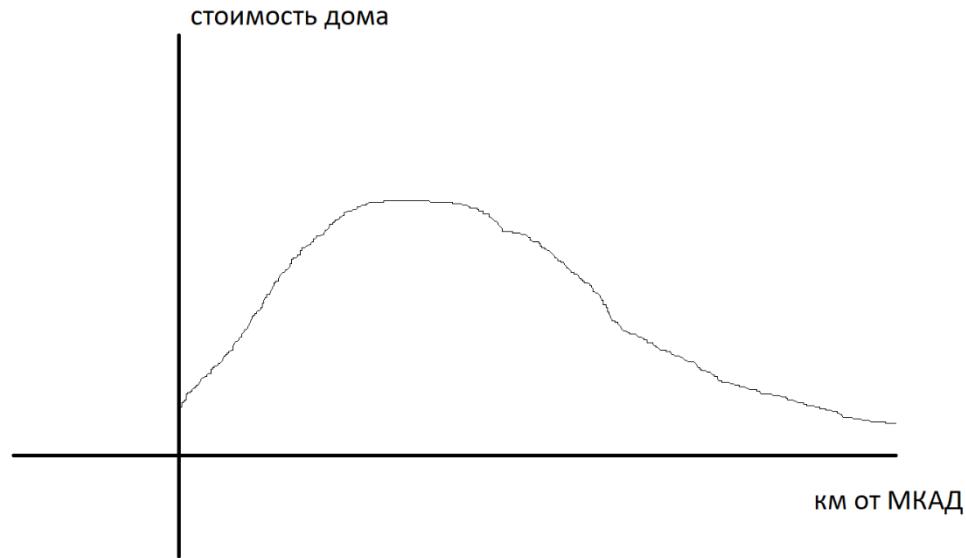
О ПРИМЕНИМОСТИ ЛИНЕЙНОЙ МОДЕЛИ

Пример:

Предположим, что мы хотим предсказать *стоимость дома* у по его *площади (x_1)* и *количество 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.

- 1) подешеветь не районов
- 2) нормировать



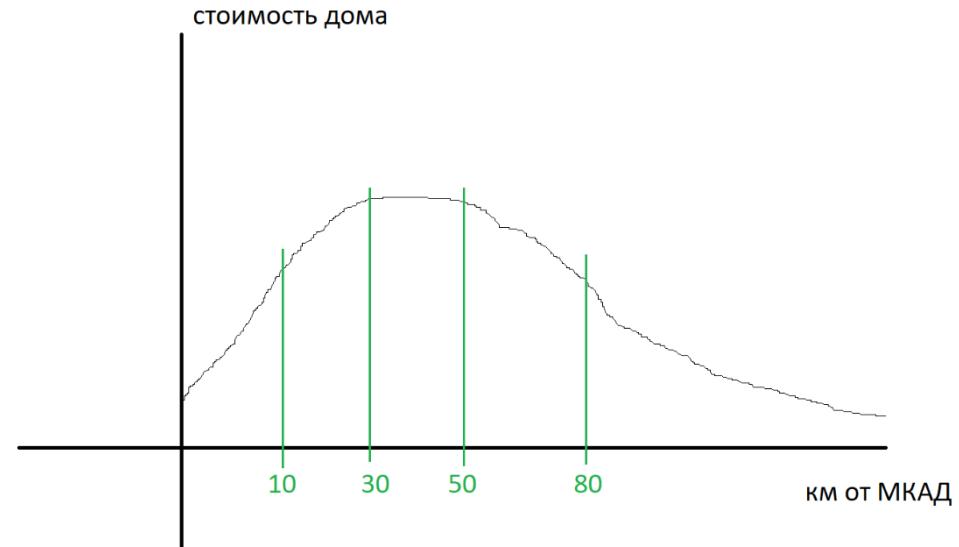
О ПРИМЕНИМОСТИ ЛИНЕЙНОЙ МОДЕЛИ

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:

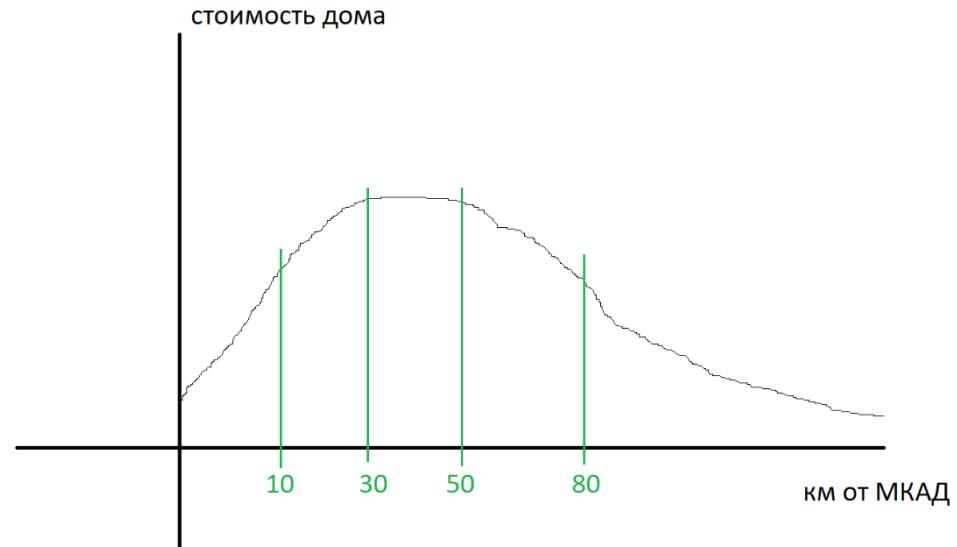
- $x_{[0;10)}$ - равен 1, если

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.



Коэффициенты: 0, 1, 2, 3, 4
Хуже: 5

$$a(x) =$$

$$= w_0 + w_1 x_1 + w_2 x_2 + \dots + w_{41} x_{[0;10)} + w_{42} x_{[10;30)} + w_{43} x_{[30;50)} + w_{44} x_{\geq 50}$$

АНАЛИТИЧЕСКОЕ РЕШЕНИЕ ЗАДАЧИ МЕТОДА НАИМЕНЬШИХ КВАДРАТОВ (МНК)

Задача обучения линейной регрессии (в матричной форме):

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

Handwritten notes in purple ink:

- $X^T X = 0$
- $X = 0$
- $y = 0$

Точное (аналитическое) решение:

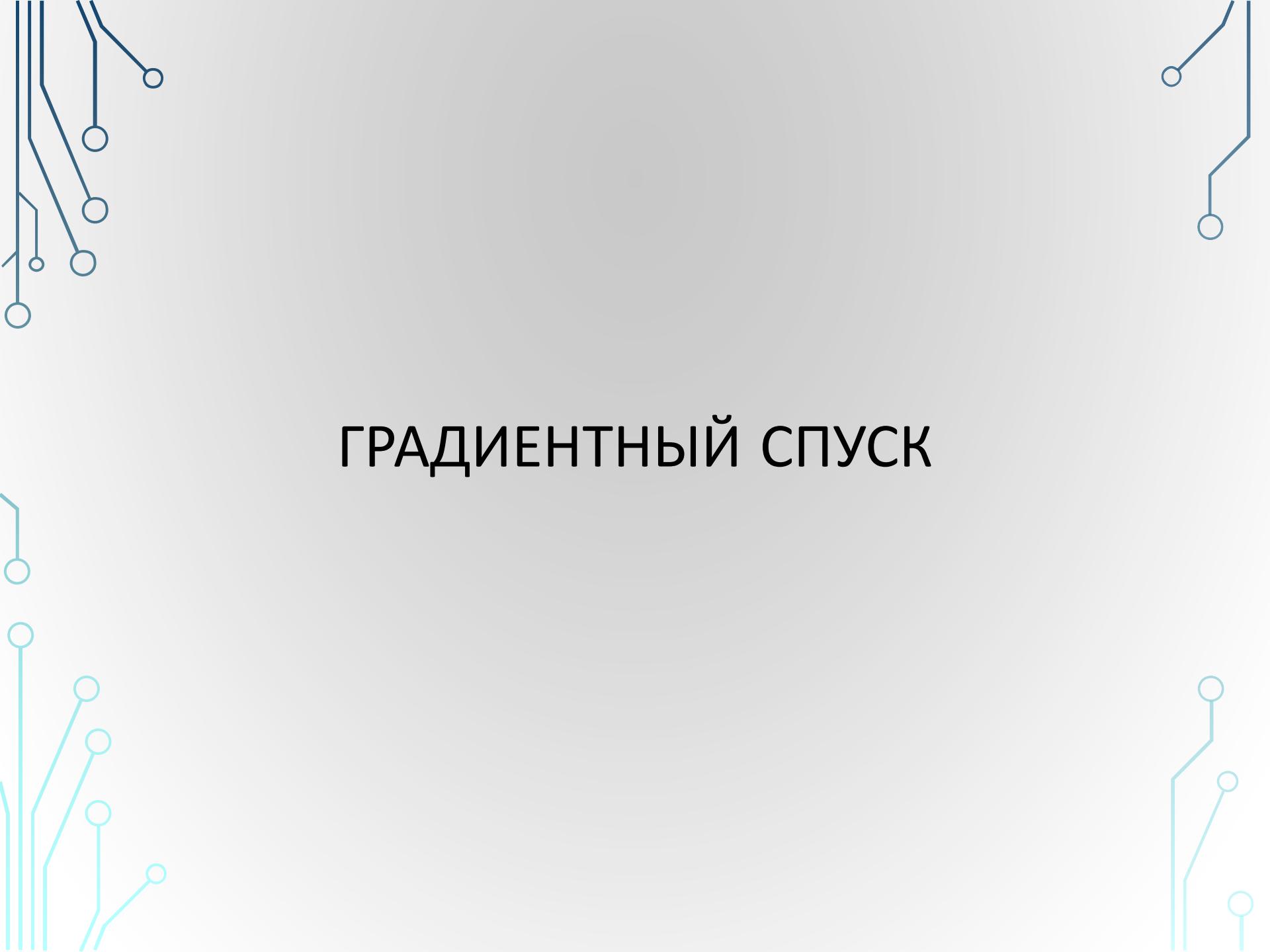
$$w = (X^T X)^{-1} X^T y$$

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

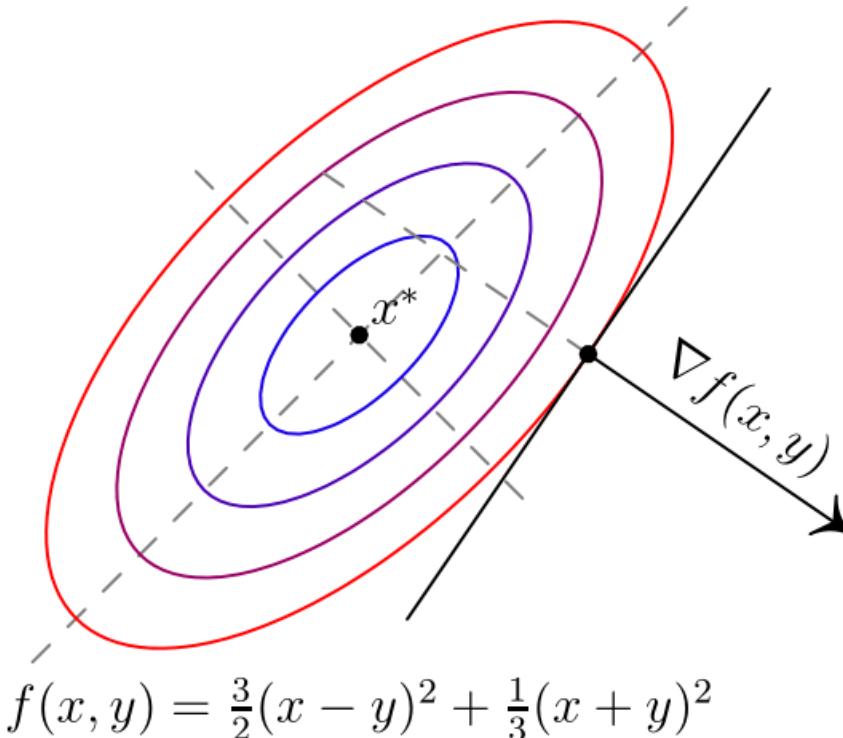
ГРАДИЕНТНЫЙ СПУСК



ТЕОРЕМА О ГРАДИЕНТЕ

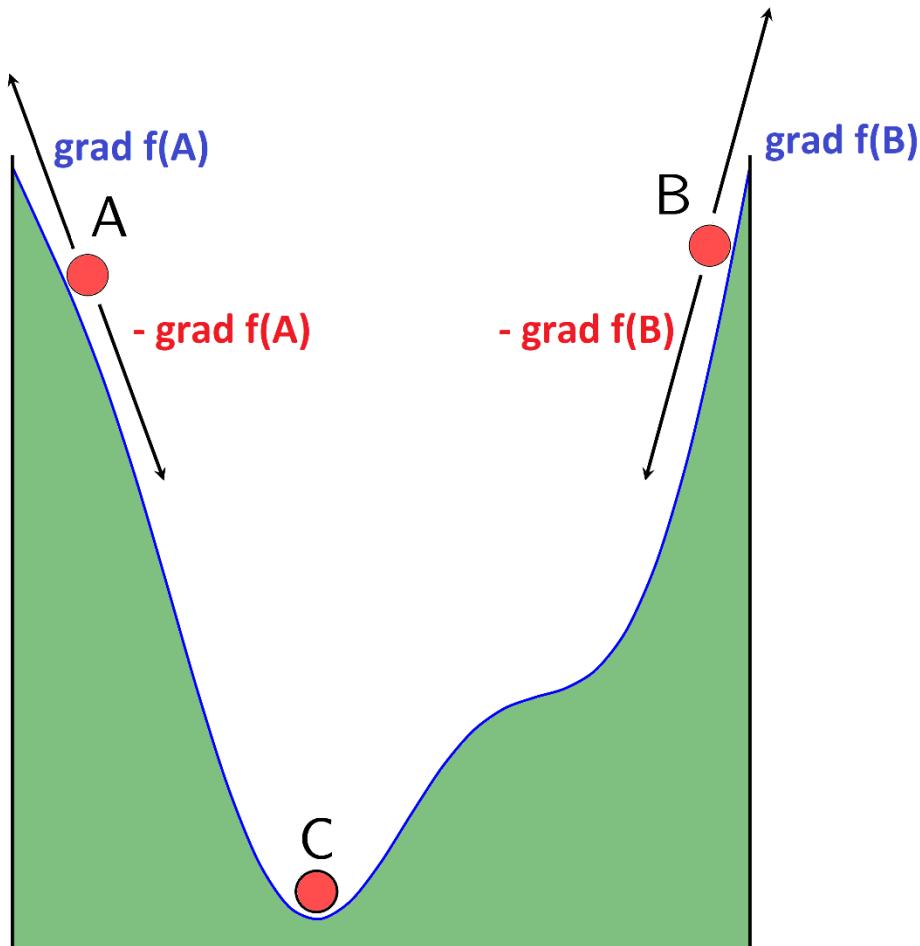
Теорема. Градиент – это вектор, в направлении которого функция быстрее всего растёт.

Антиградиент (вектор, противоположный градиенту) – вектор, в направлении которого функция быстрее всего убывает.



ТЕОРЕМА О ГРАДИЕНТЕ

Антиградиент (вектор, противоположный градиенту) – вектор, в направлении которого функция быстрее всего убывает.

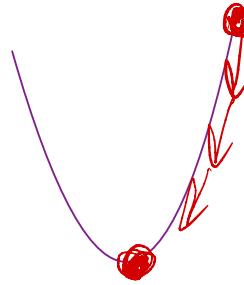


МЕТОД ГРАДИЕНТНОГО СПУСКА

- 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.

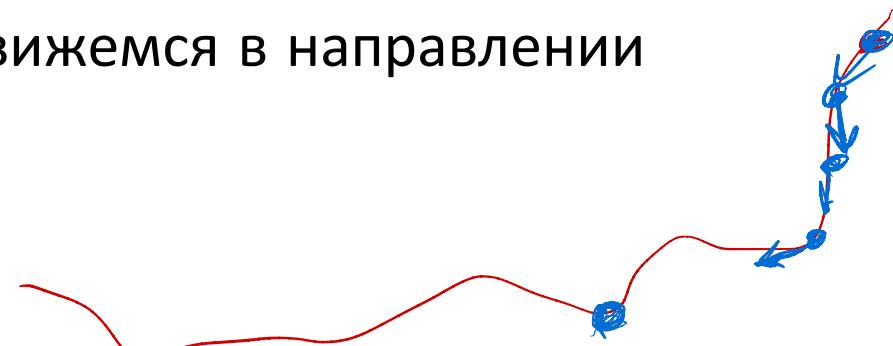


МЕТОД ГРАДИЕНТНОГО СПУСКА

- 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:

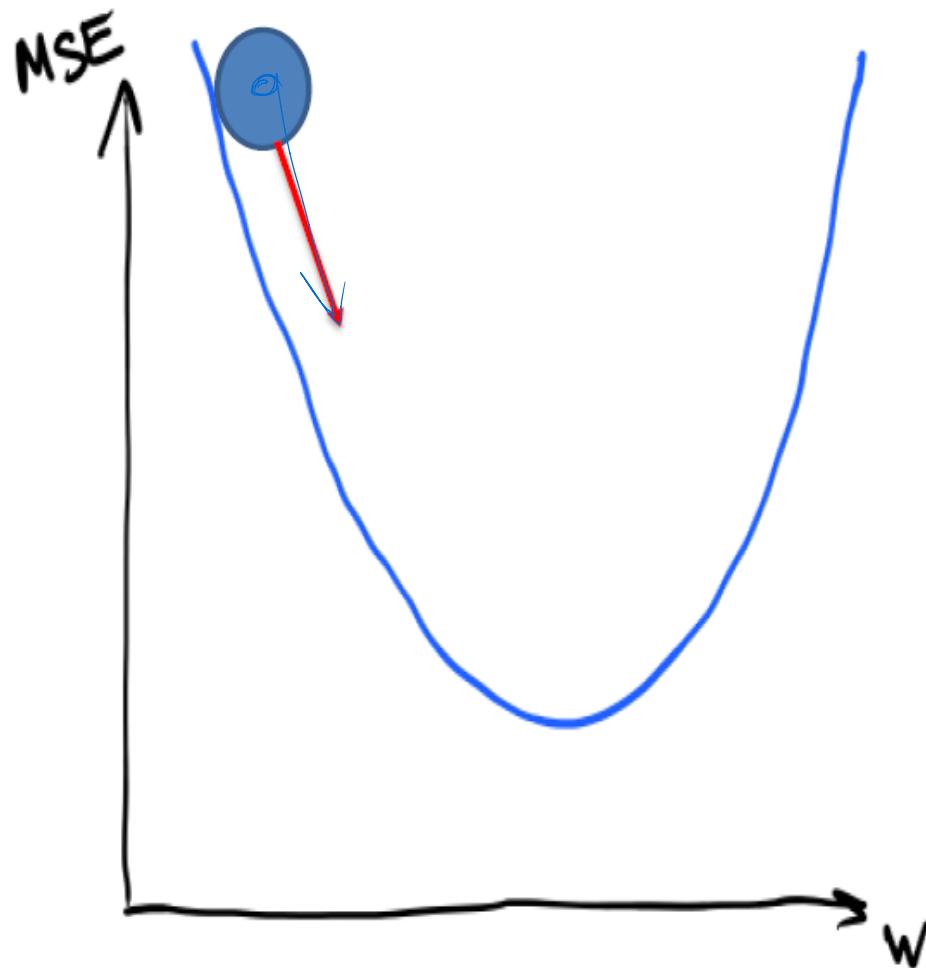
На каждом шаге (на каждой итерации метода) движемся в сторону антиградиента функции потерь!

То есть на каждом шаге движемся в направлении уменьшения ошибки.

Вектор градиента функции потерь обозначают *grad Q* или ∇Q .

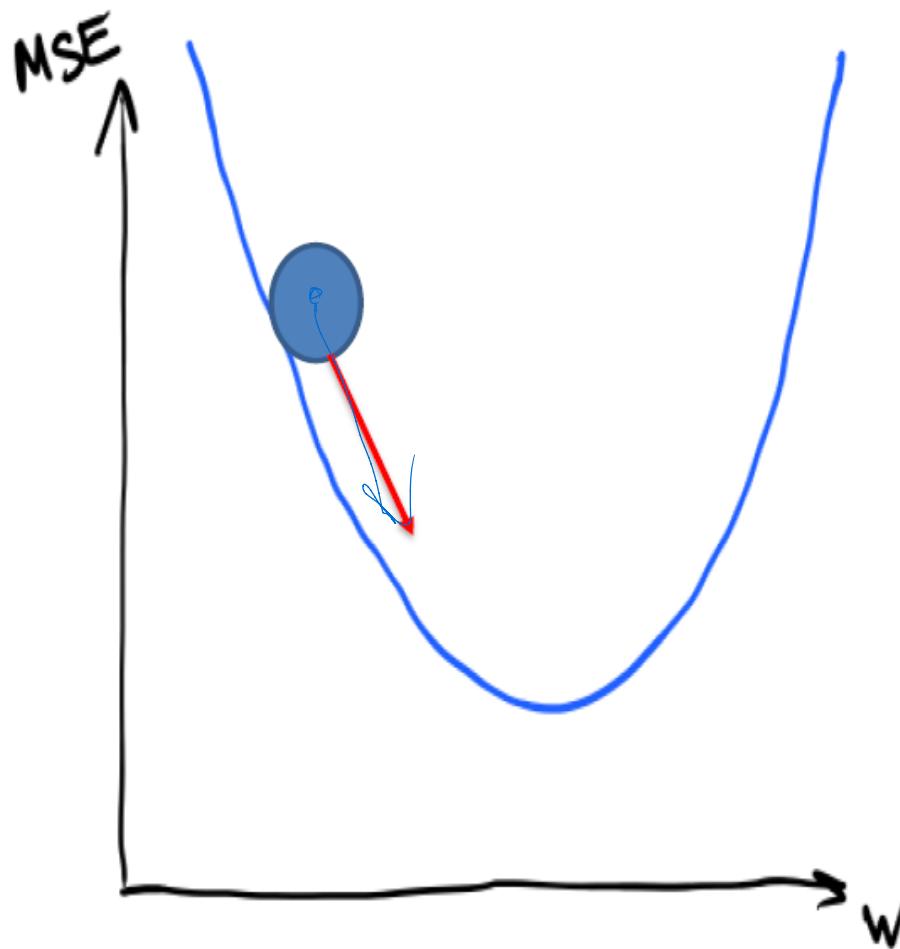
МЕТОД ГРАДИЕНТНОГО СПУСКА

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



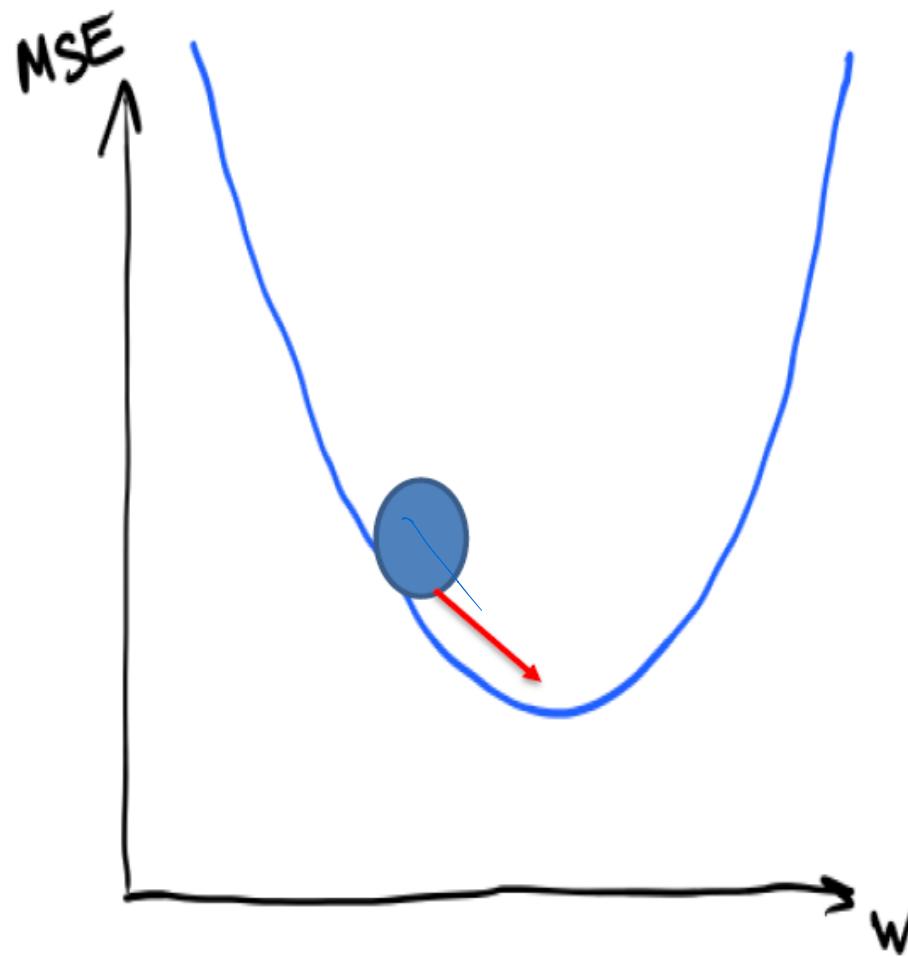
МЕТОД ГРАДИЕНТНОГО СПУСКА

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



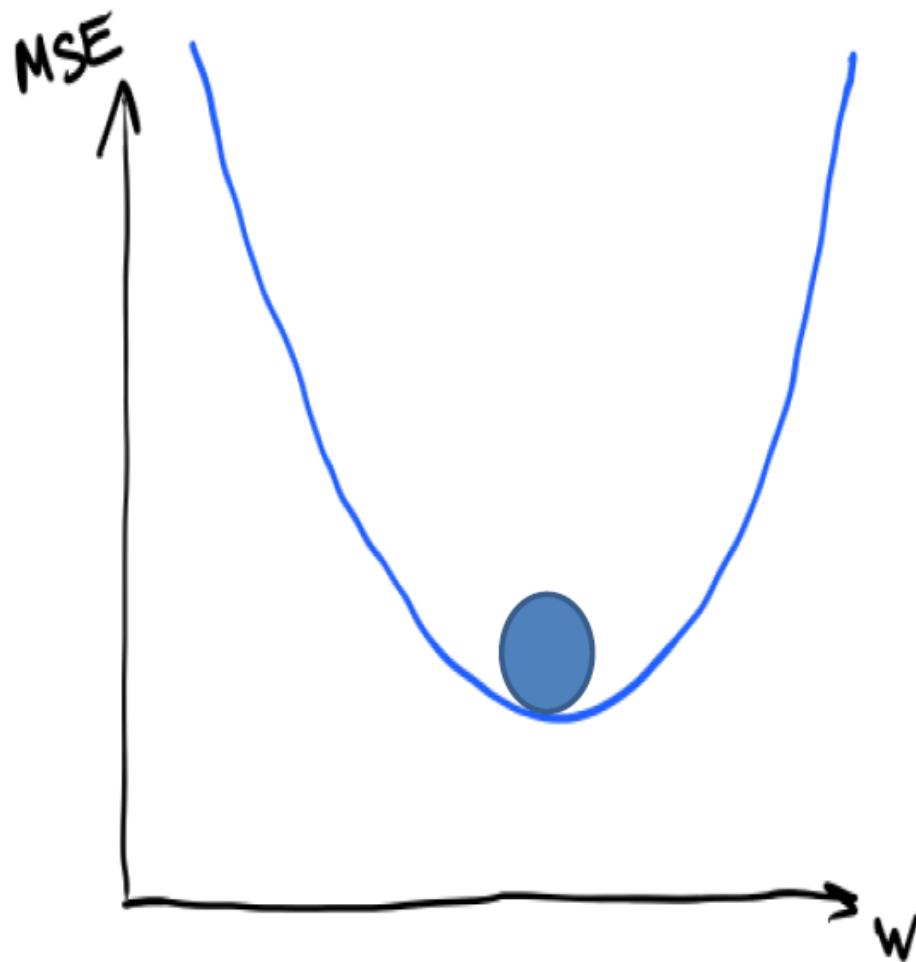
МЕТОД ГРАДИЕНТНОГО СПУСКА

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

$$y = w^T x$$

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 .

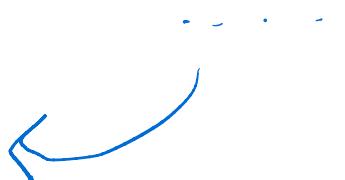
Тогда при добавлении к весу w слагаемого $-\frac{\partial Q}{\partial w}$ функция $Q(w)$ убывает.

- Инициализируем вес $w^{(0)}$.
- На каждом следующем шаге обновляем вес, добавляя

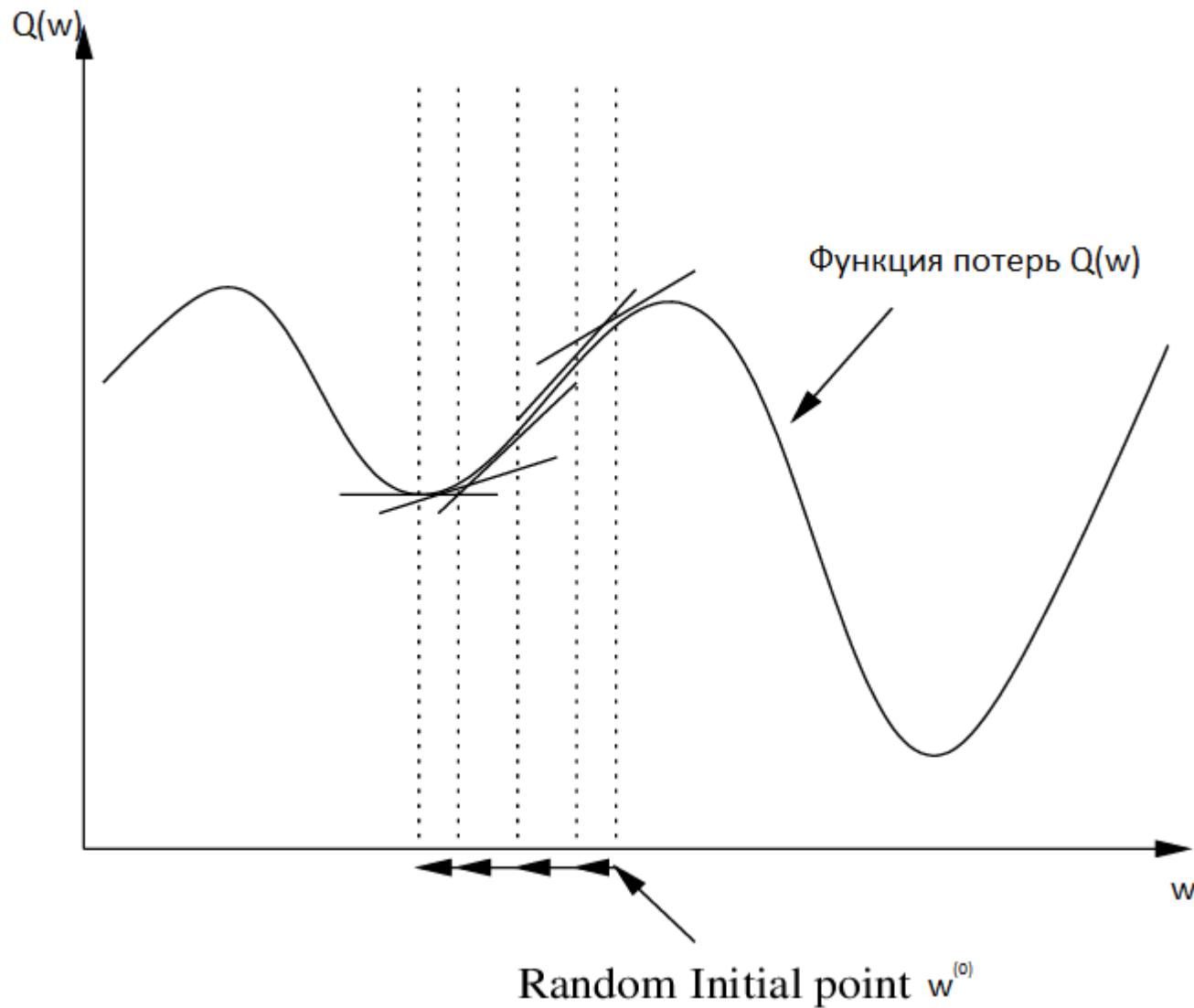
$$-\frac{\partial Q}{\partial w}(w^{(k-1)}) :$$

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

$$w^{(2)} = w^{(1)} - \frac{\partial Q}{\partial w}(w^{(1)})$$



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

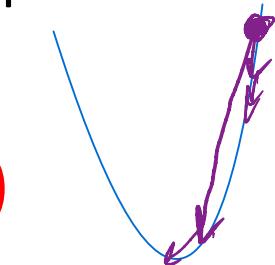
Формулу для обновления весов можно записать в векторном виде:

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

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

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

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



МЕТОД ГРАДИЕНТНОГО СПУСКА

Формулу для обновления весов можно записать в векторном виде:

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

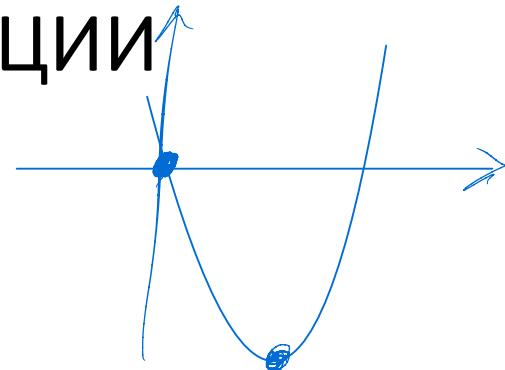
$$w^{(k)} = w^{(k-1)} - \nabla Q(w^{(k-1)})$$

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

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

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

ВАРИАНТЫ ИНИЦИАЛИЗАЦИИ ВЕСОВ

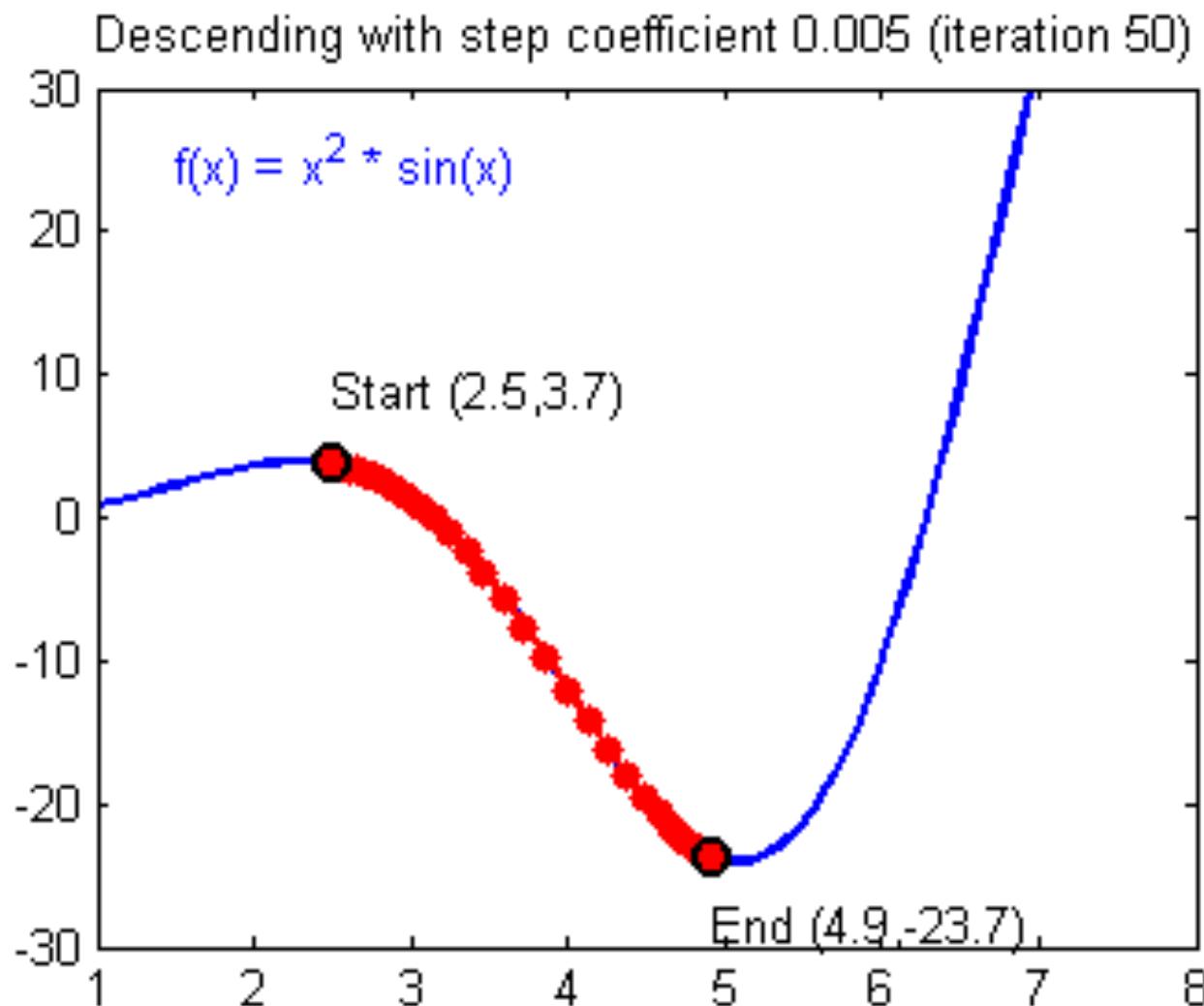


- $w_j = 0, j = 1, \dots, n$
- Небольшие случайные значения:
$$w_j := \text{random}(-\varepsilon, \varepsilon)$$
- Обучение по небольшой случайной подвыборке объектов
- Мультистарт: многократный запуск из разных случайных начальных приближений и выбор лучшего решения

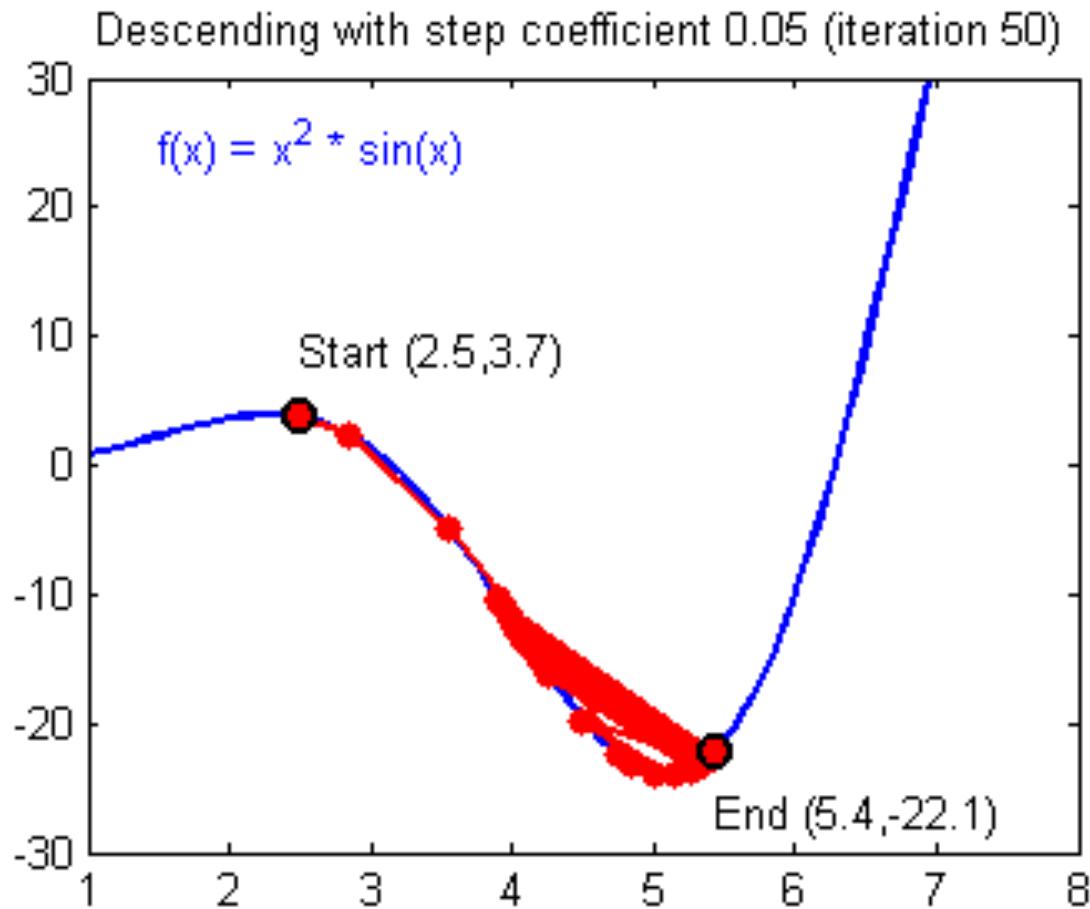
КРИТЕРИИ ОСТАНОВА

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

ГРАДИЕНТНЫЙ СПУСК



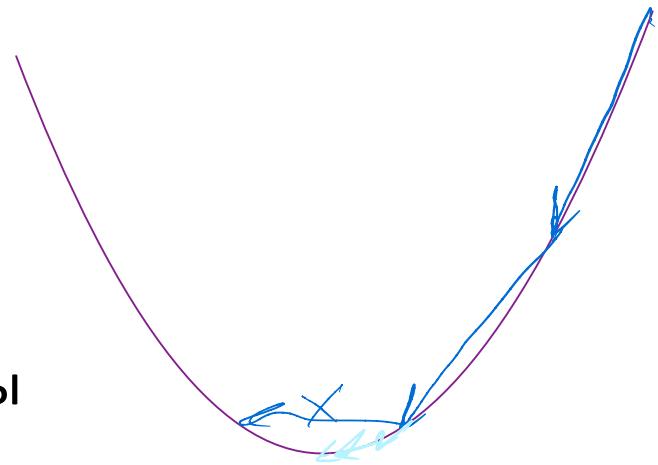
ПРОБЛЕМА ВЫБОРА ГРАДИЕНТНОГО ШАГА



ГРАДИЕНТНЫЙ ШАГ

В общем случае градиентный шаг может зависеть от номера итерации, тогда будем писать не η , а η_k .

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

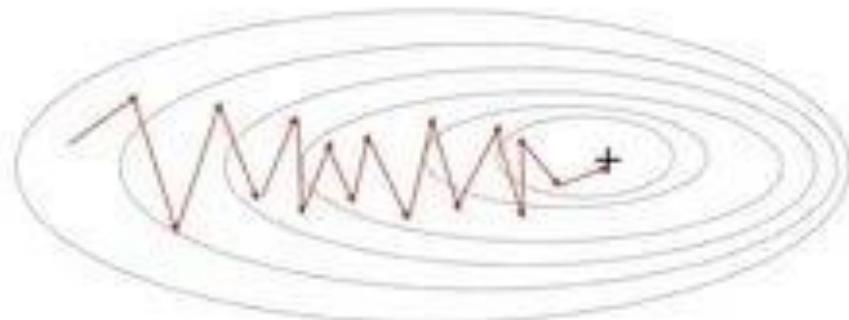
- на каждом шаге выбираем **один случайный объект** и сдвигаемся в сторону антиградиента по этому объекту:

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

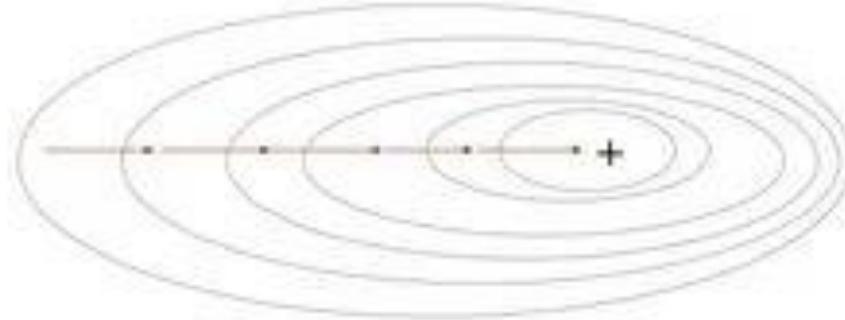
где $\nabla q_{i_k}(w^{(k-1)})$ - градиент функции потерь, вычисленный только по объекту с номером i_k (а не по всей обучающей выборке).

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 η через некоторое число шагов гарантированно попадет в малую окрестность точки w^* . Однако, сходится метод медленнее, чем обычный градиентный спуск

MINI-BATCH GRADIENT DESCENT

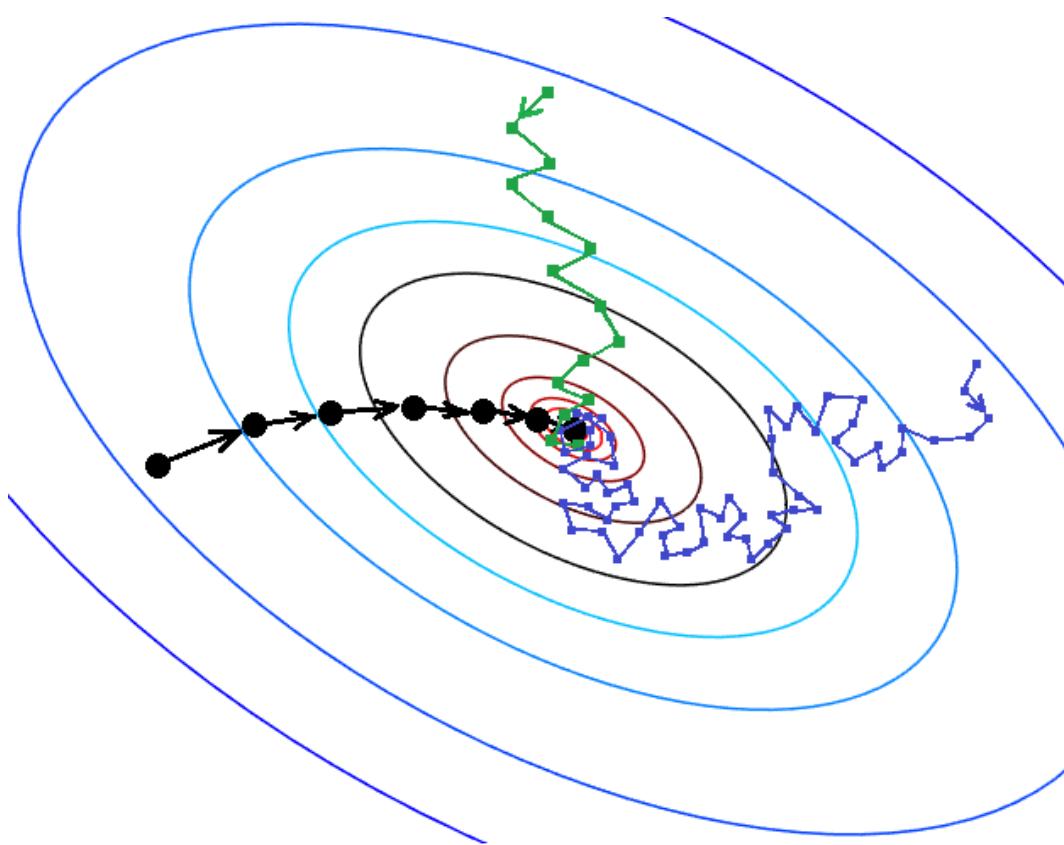
Промежуточное решение между классическим градиентным спуском и стохастическим вариантом.

- 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

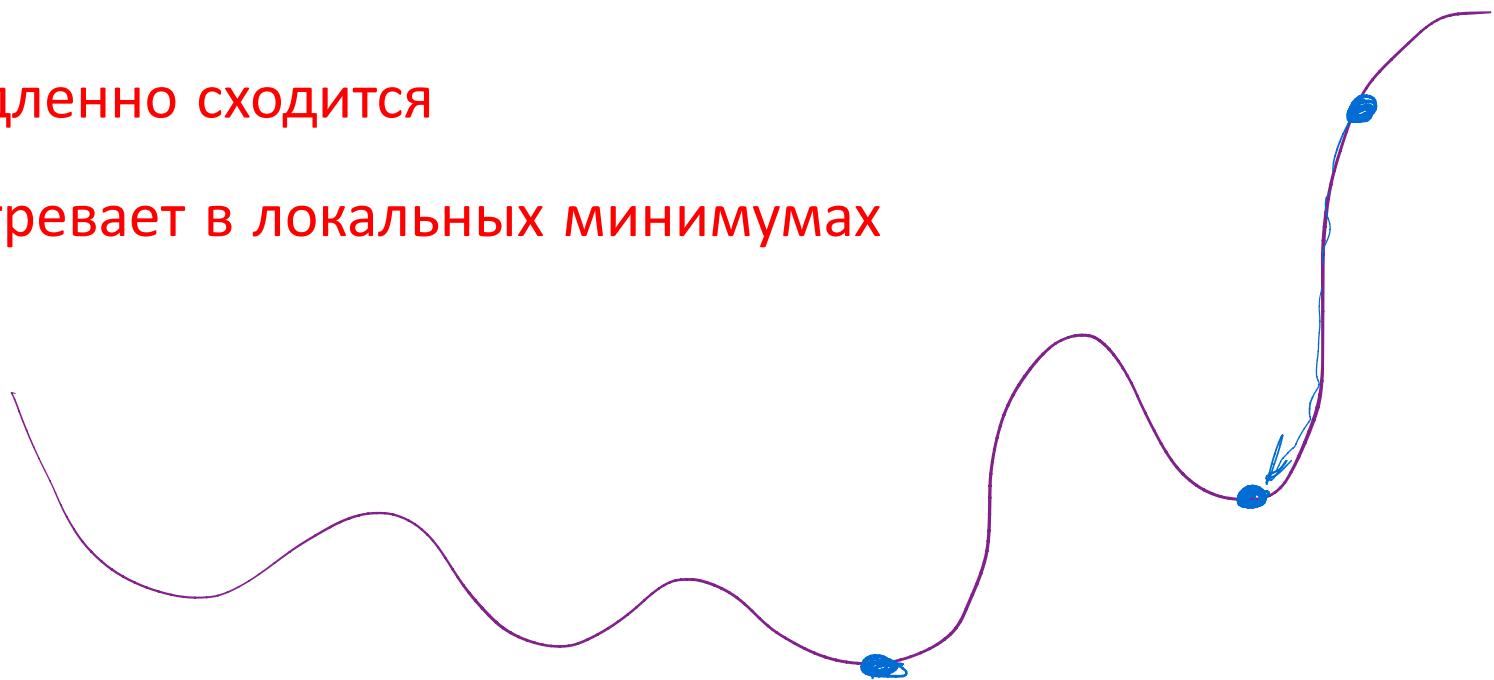
Mini-batch GD

- Compromise

МОДИФИКАЦИИ ГРАДИЕНТНОГО СПУСКА

БОНУС: ПРОБЛЕМЫ ГРАДИЕНТНОГО СПУСКА И ВАРИАНТЫ ИХ РЕШЕНИЯ

- Медленно сходится
- Застревает в локальных минимумах



ПРОБЛЕМА ЗАСТРЕВАНИЯ В LOCMIN



МЕТОД МОМЕНТОВ (МОМЕНТУМ)

Вектор инерции (*усреднение градиента по предыдущим шагам*):

$$h_0 = 0$$

$$h_k = \alpha h_{k-1} + \eta_k \nabla Q(w^{(k-1)})$$

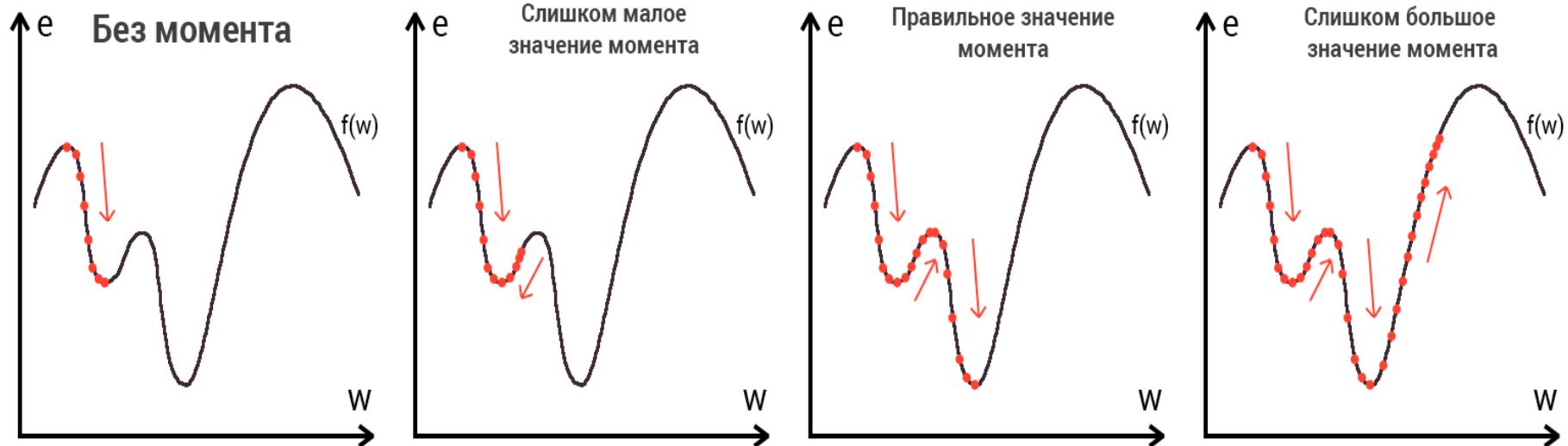
Формула метода моментов:

$$w^{(k)} = w^{(k-1)} - h_k$$

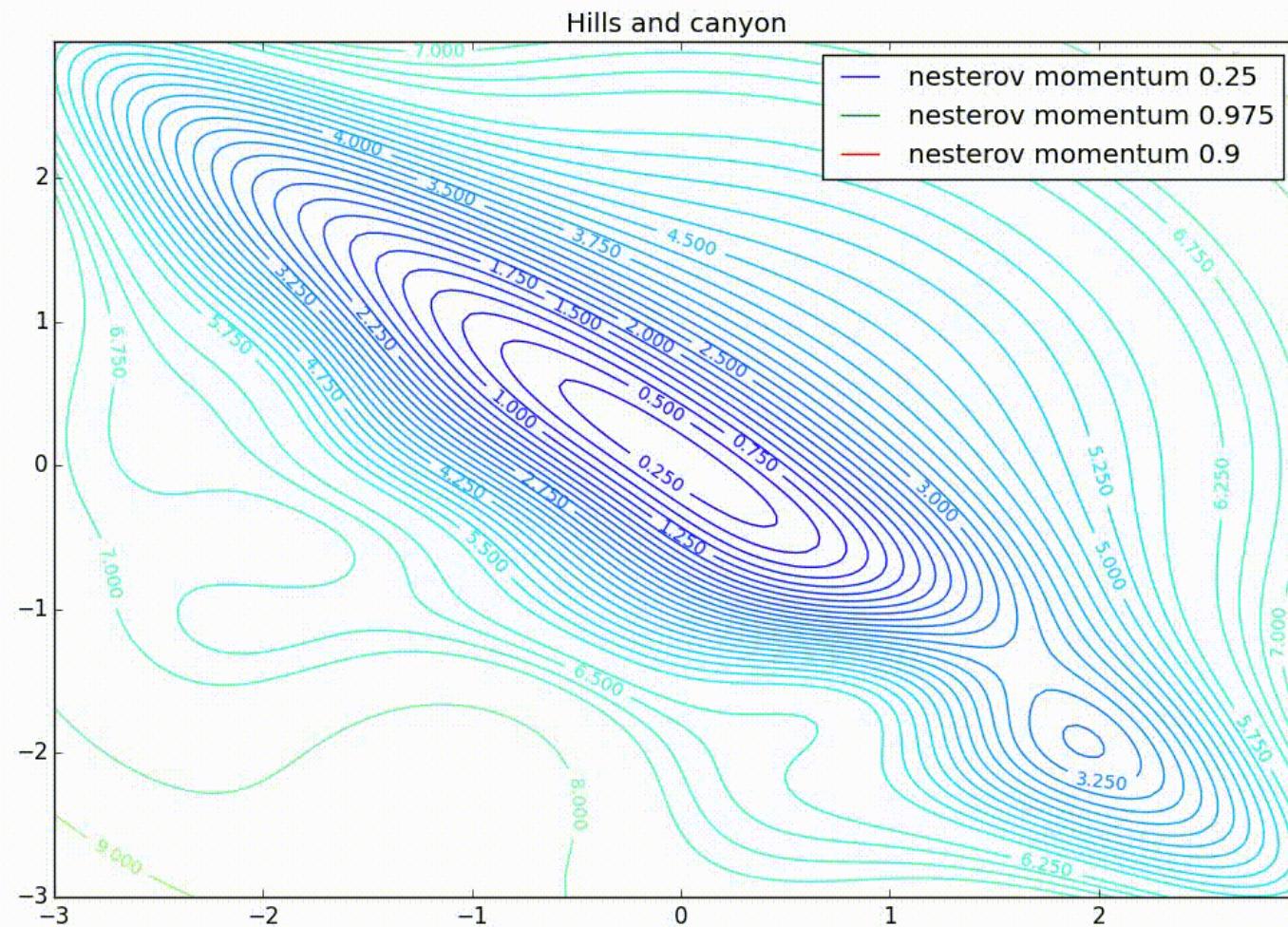
Подробнее:

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

MOMENTUM



MOMENTUM



ADAGRAD (ADAPTIVE GRADIENT)

Сумма квадратов обновлений:

$$g_{k-1,j} = (\nabla Q(w^{(k-1)}))_j^2$$

Формулы метода AdaGrad:

- $G_{k,j} = G_{k-1,j} + g_{k-1,j} = G_{k-1,j} + (\nabla Q(w^{(k-1)}))_j^2$
- $\omega_j^{(k)} = \omega_j^{k-1} - \frac{\eta}{\sqrt{G_{k,j} + \epsilon}} \cdot (\nabla Q(w^{(k-1)}))_j$

*Этот метод использует адаптивный шаг обучения
– тем самым мы регулируем скорость сходимости
метода.*

ADAGRAD (ADAPTIVE GRADIENT)

Сумма квадратов обновлений:

$$g_{k-1,j} = (\nabla Q(w^{(k-1)}))_j^2$$

Формулы метода AdaGrad:

- $G_{k,j} = G_{k-1,j} + g_{k-1,j}$
 - $\omega_j^{(k)} = \omega_j^{k-1} - \frac{\eta}{\sqrt{G_{k,j} + \varepsilon}} \cdot (\nabla Q(w^{(k-1)}))_j$
- + Автоматическое затухание скорости обучения
- G_{kj} монотонно возрастают, поэтому шаги укорачиваются, и мы можем не успеть дойти до минимума

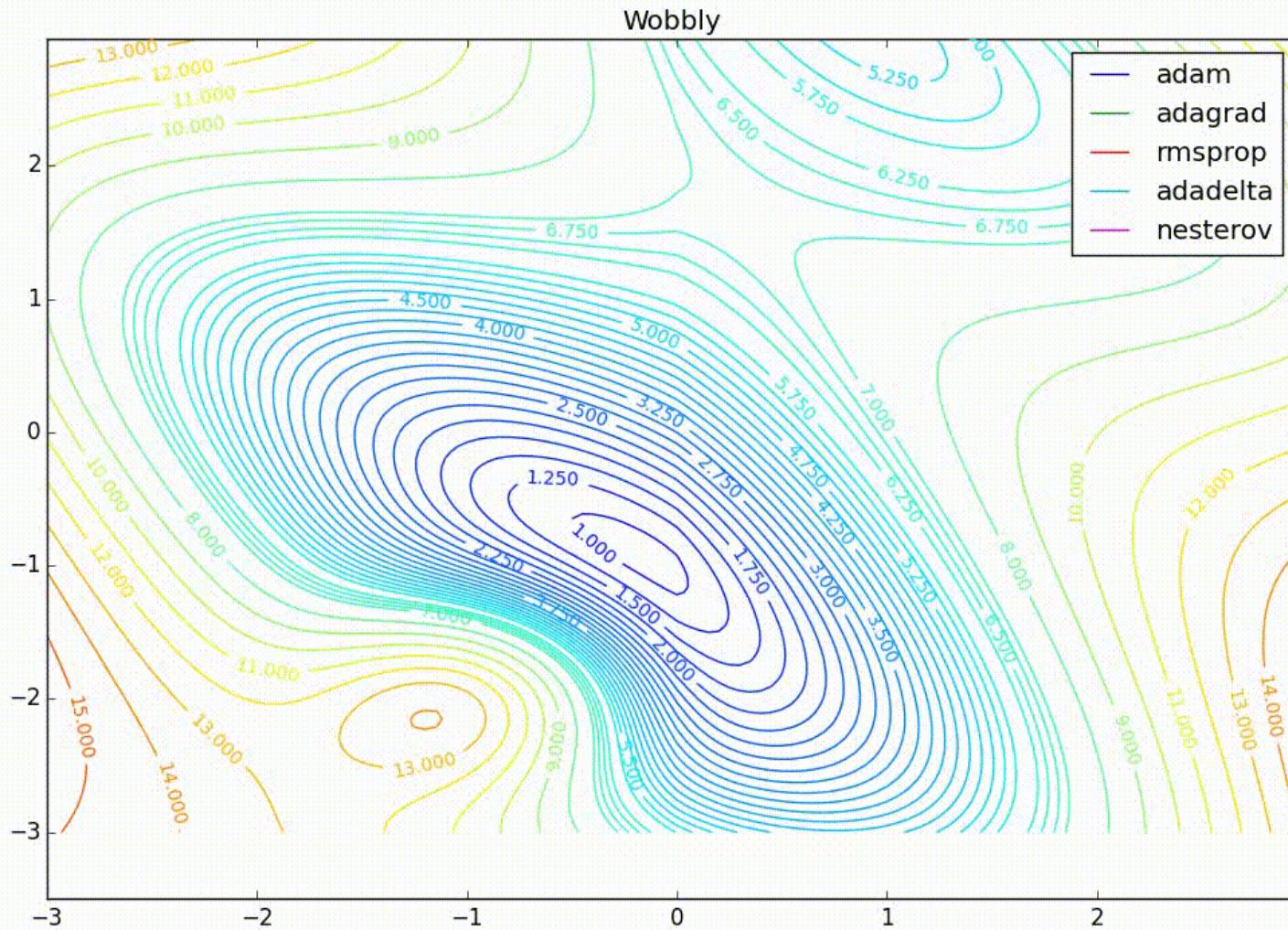
RMSPROP (ROOT MEAN SQUARE PROPAGATION)

Метод реализует экспоненциальное затухание градиентов

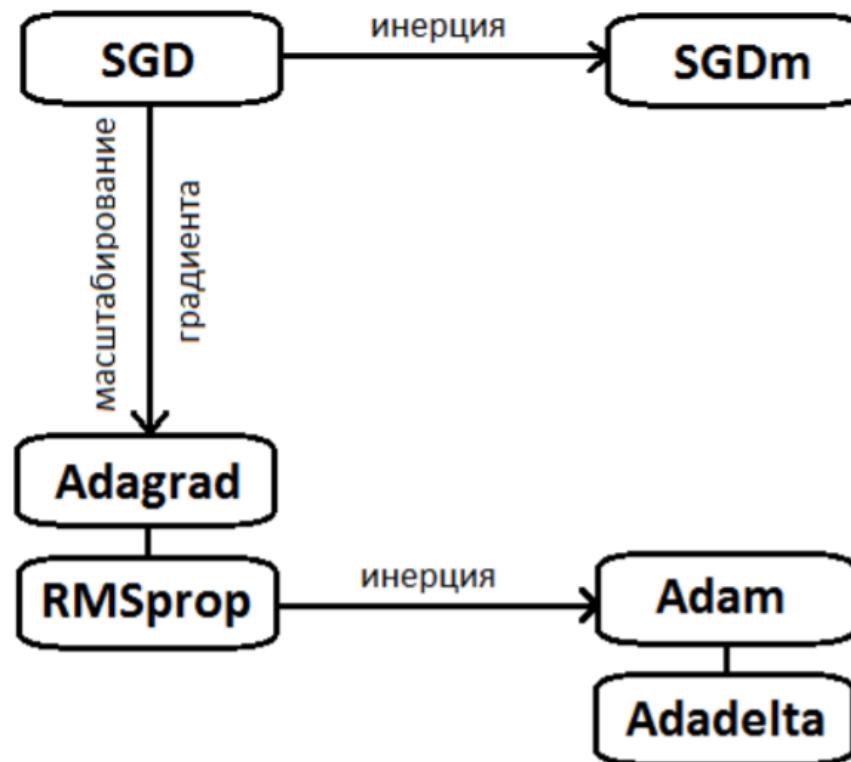
Формулы метода RMSprop (*усредненный по истории квадрат градиента*):

- $G_{k,j} = \alpha \cdot G_{k-1,j} + (1 - \alpha) \cdot g_{k-1,j}$
- $\omega_j^{(k)} = \omega_j^{k-1} - \frac{\eta}{\sqrt{G_{k,j} + \varepsilon}} \cdot \left(\nabla Q(w^{(k-1)}) \right)_j$

МОДИФИКАЦИИ ГРАДИЕНТНОГО СПУСКА



МОДИФИКАЦИИ SGD



[ссылка на статью](#)