



# Занятие 2

# Линейные методы

# регрессии. Часть 1.

Елена Кантоностова

[elena.kantonistova@yandex.ru](mailto:elena.kantonistova@yandex.ru)

ВШЭ, 2020

# 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).



# ЛИНЕЙНАЯ РЕГРЕССИЯ

Пример (напоминание):

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

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:

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

- сокращенная запись:

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

# ЛИНЕЙНАЯ РЕГРЕССИЯ

Линейная регрессия:

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

- 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:

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

- 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)

# МЕТОД МАКСИМУМА ПРАВДОПОДОБИЯ

Модель данных с некоррелированным гауссовским шумом:

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

# МЕТОД МАКСИМУМА ПРАВДОПОДОБИЯ

Модель данных с некоррелированным гауссовским шумом:

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

Метод максимума правдоподобия (ММП):

$$L(\varepsilon_1, \dots, \varepsilon_l | w) = \prod_{i=1}^l \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2} \varepsilon_i^2\right) \rightarrow \max_w$$

# МЕТОД МАКСИМУМА ПРАВДОПОДОБИЯ

Модель данных с некоррелированным гауссовским шумом:

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

Метод максимума правдоподобия (ММП):

$$L(\varepsilon_1, \dots, \varepsilon_l | w) = \prod_{i=1}^l \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2} \varepsilon_i^2\right) \rightarrow \max_w$$

$$-\ln L(\varepsilon_1, \dots, \varepsilon_l | w) = \text{const} + \frac{1}{2\sigma^2} \sum_{i=1}^l ((w, x_i) - y_i)^2 \rightarrow \min_w$$

В данном случае ММП совпадает с МНК.

# 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:

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

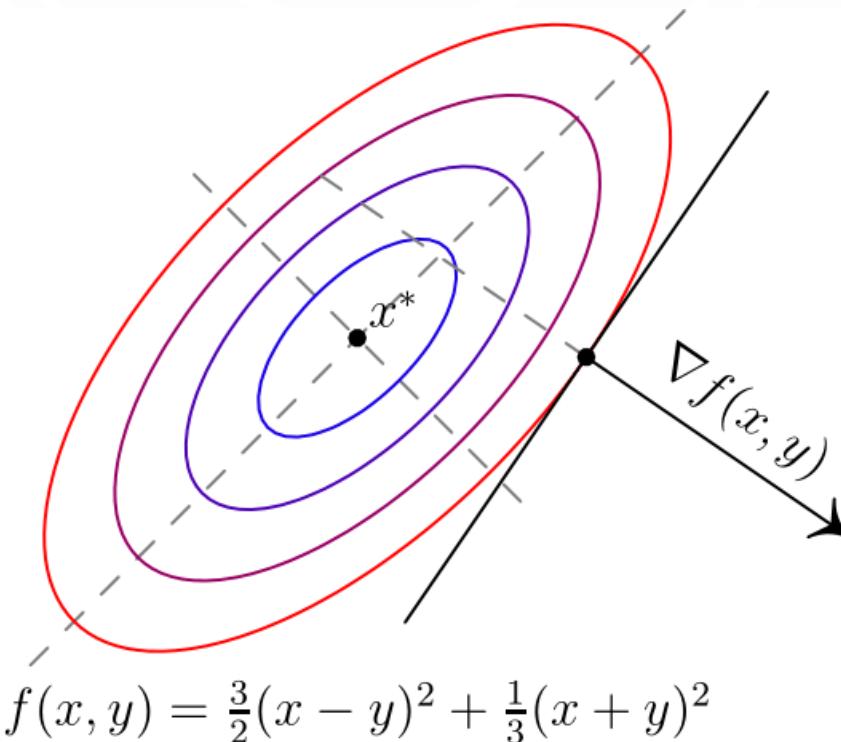
# НЕДОСТАТКИ АНАЛИТИЧЕСКОЙ ФОРМУЛЫ

- 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

# ТЕОРЕМА О ГРАДИЕНТЕ

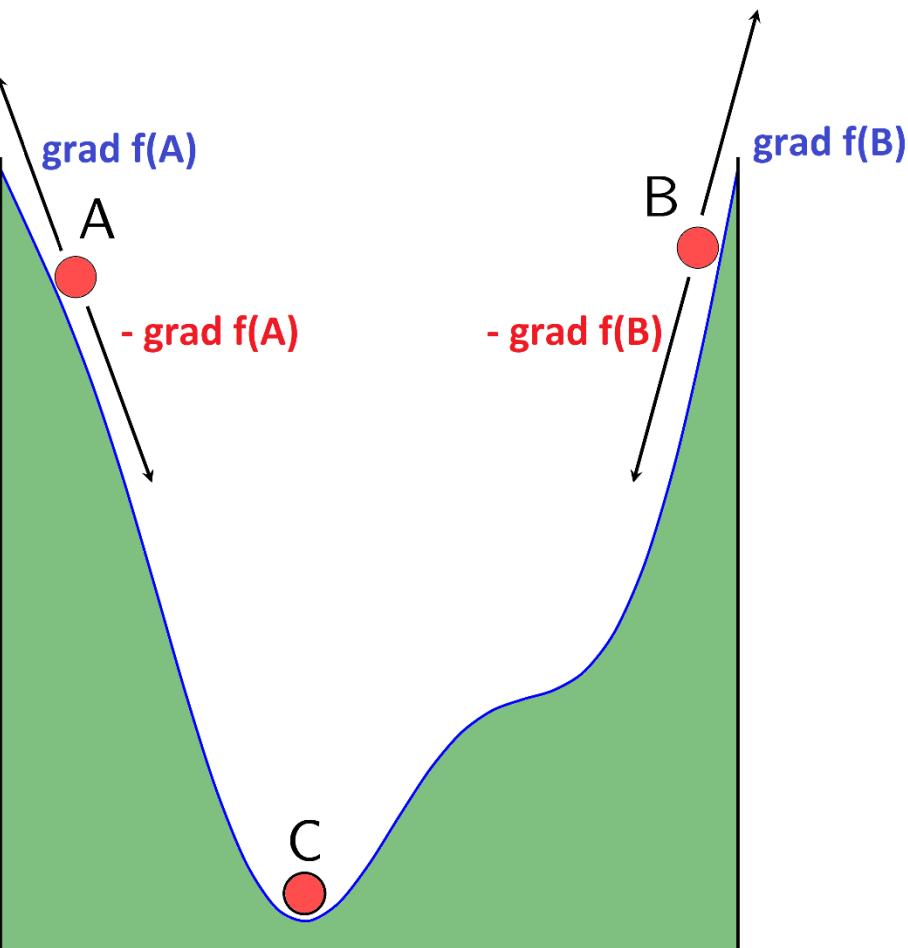
**Теорема.** Градиент – это вектор, в направлении которого функция быстрее всего растёт.

**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 график среднеквадратичной ошибки – это парабола.

# МЕТОД ГРАДИЕНТНОГО СПУСКА

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

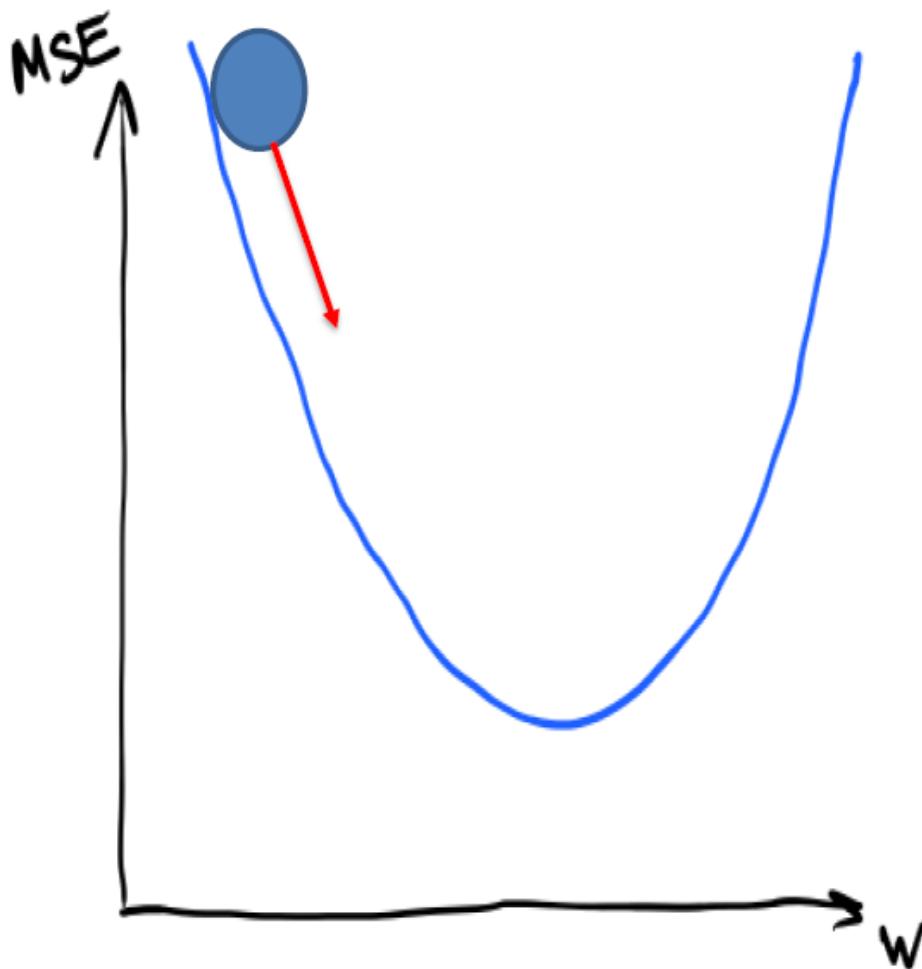
На каждом шаге (на каждой итерации метода) движемся в сторону антиградиента функции потерь!

То есть на каждом шаге движемся в направлении уменьшения ошибки.

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

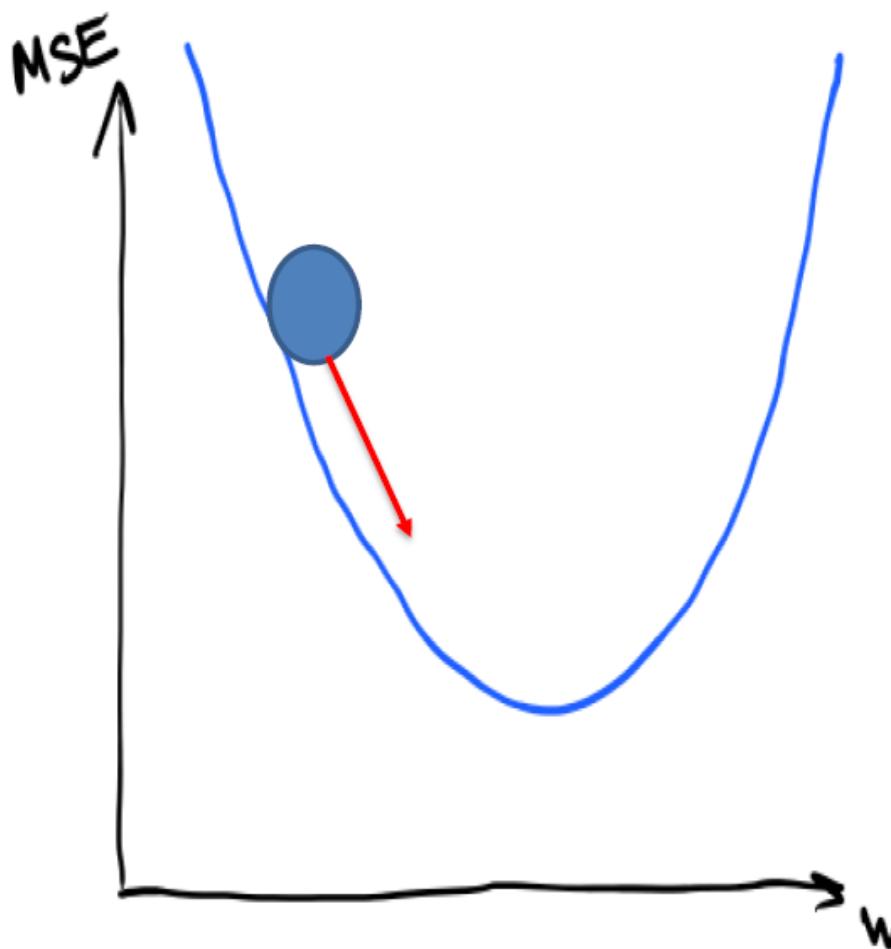
# МЕТОД ГРАДИЕНТНОГО СПУСКА

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



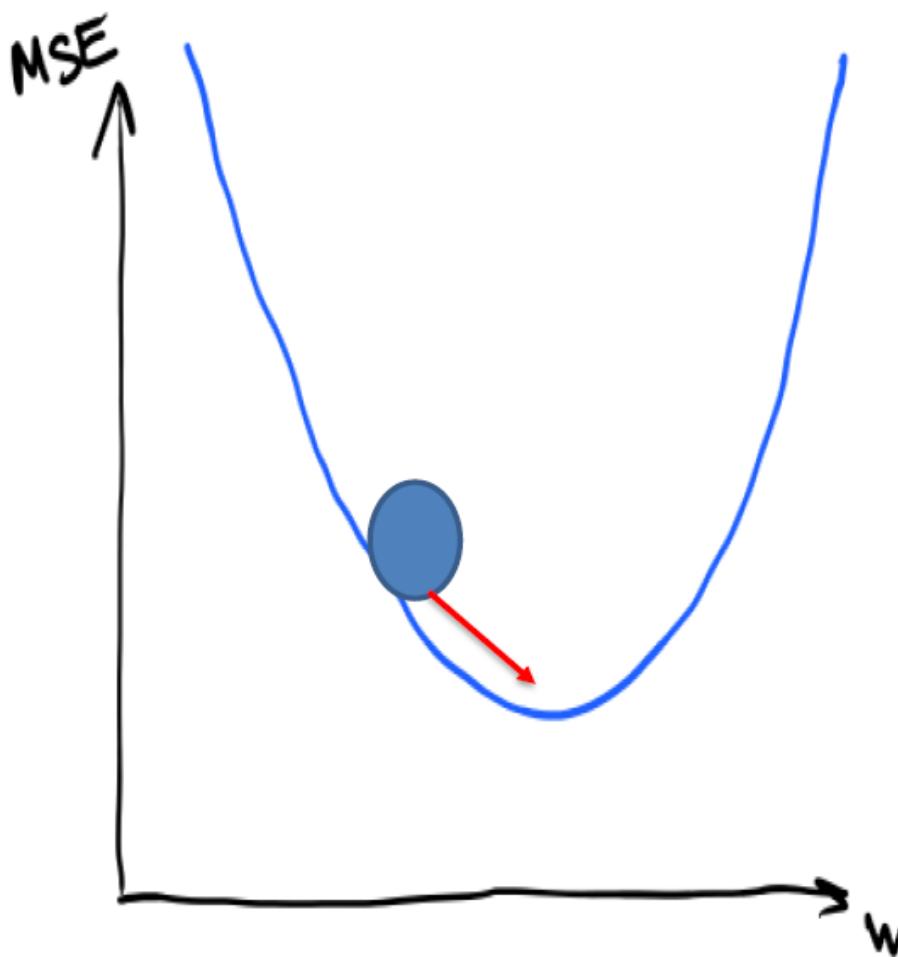
# МЕТОД ГРАДИЕНТНОГО СПУСКА

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



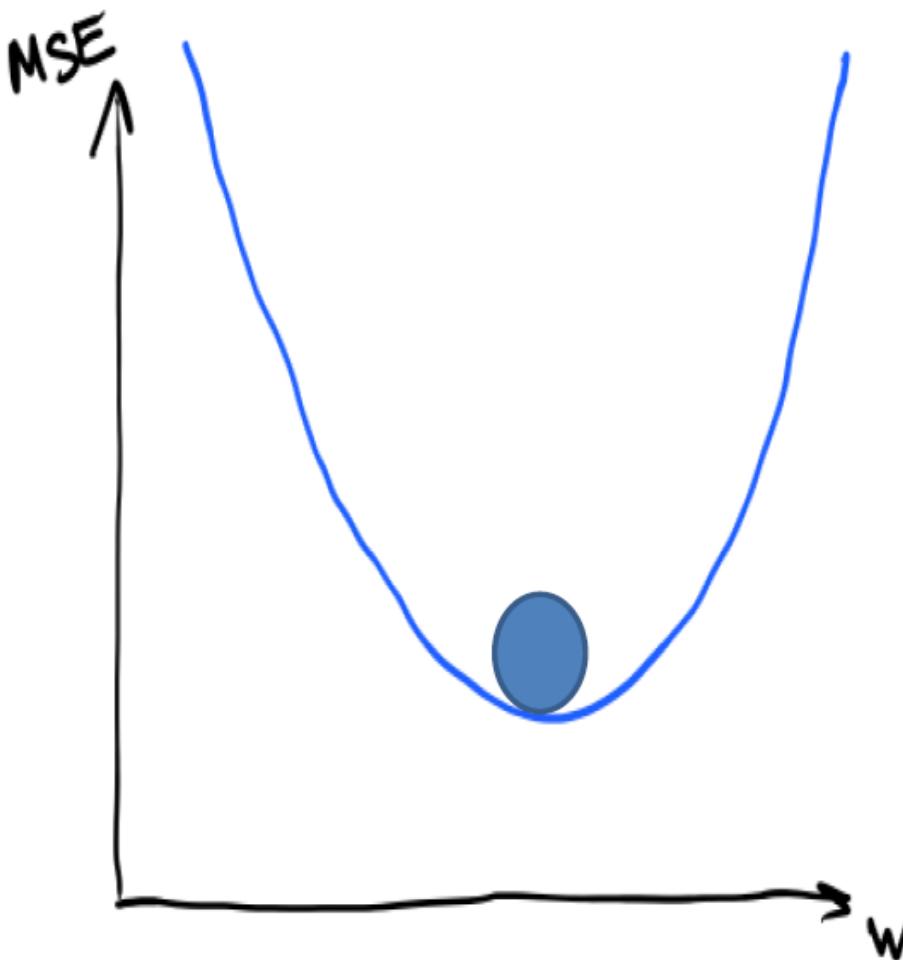
# МЕТОД ГРАДИЕНТНОГО СПУСКА

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

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

**Метод градиентного спуска:**

- Инициализируем веса  $w_0^{(0)}, w_1^{(0)}, w_2^{(0)}, \dots, w_n^{(0)}$ .

# МЕТОД ГРАДИЕНТНОГО СПУСКА

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

**Метод градиентного спуска:**

- Инициализируем веса  $w_0^{(0)}, w_1^{(0)}, w_2^{(0)}, \dots, w_n^{(0)}$ .
- На каждом следующем шаге обновляем веса, сдвигаясь в направлении антиградиента функции потерь  $Q$ :

$$w_0^{(k)} = w_0^{(k-1)} - \nabla Q(w_0^{(k-1)}),$$

# МЕТОД ГРАДИЕНТНОГО СПУСКА

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

**Метод градиентного спуска:**

- Инициализируем веса  $w_0^{(0)}, w_1^{(0)}, w_2^{(0)}, \dots, w_n^{(0)}$ .
- На каждом следующем шаге обновляем веса, сдвигаясь в направлении антиградиента функции потерь  $Q$ :

$$w_0^{(k)} = w_0^{(k-1)} - \nabla Q(w_0^{(k-1)}),$$

$$w_1^{(k)} = w_1^{(k-1)} - \nabla Q(w_1^{(k-1)}),$$

...

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

# МЕТОД ГРАДИЕНТНОГО СПУСКА

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

**Метод градиентного спуска 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

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

Метод градиентного спуска 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  $\eta$  – величина градиентного шага (learning rate). Он отвечает за скорость движения в сторону антиградиента:

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

# ГРАДИЕНТНЫЙ СПУСК

Градиент функции  $Q$  вычисляется как сумма градиентов функции потерь  $q_i(w)$  по всем объектам:

$$\nabla Q(w) = \sum_{i=1}^l \nabla q_i(w)$$

Градиентный спуск:

$$w^{(k)} = w^{(k-1)} - \eta \sum_{i=1}^l \nabla q_i(w^{(k-1)})$$

Скорость сходимости:  $Q(w^{(k)}) - Q(w^*) = O\left(\frac{1}{k}\right)$

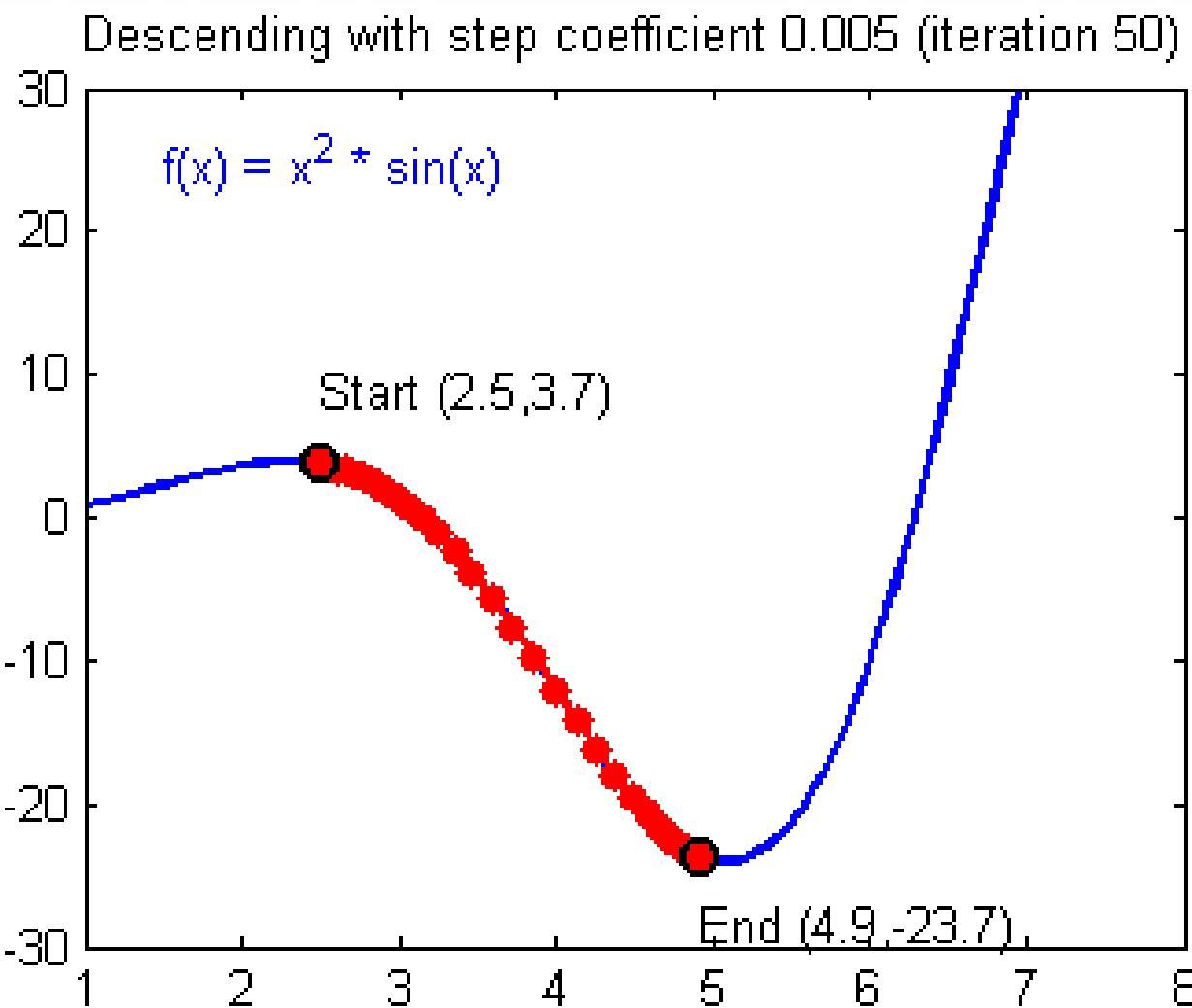
# ВАРИАНТЫ ИНИЦИАЛИЗАЦИИ ВЕСОВ

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

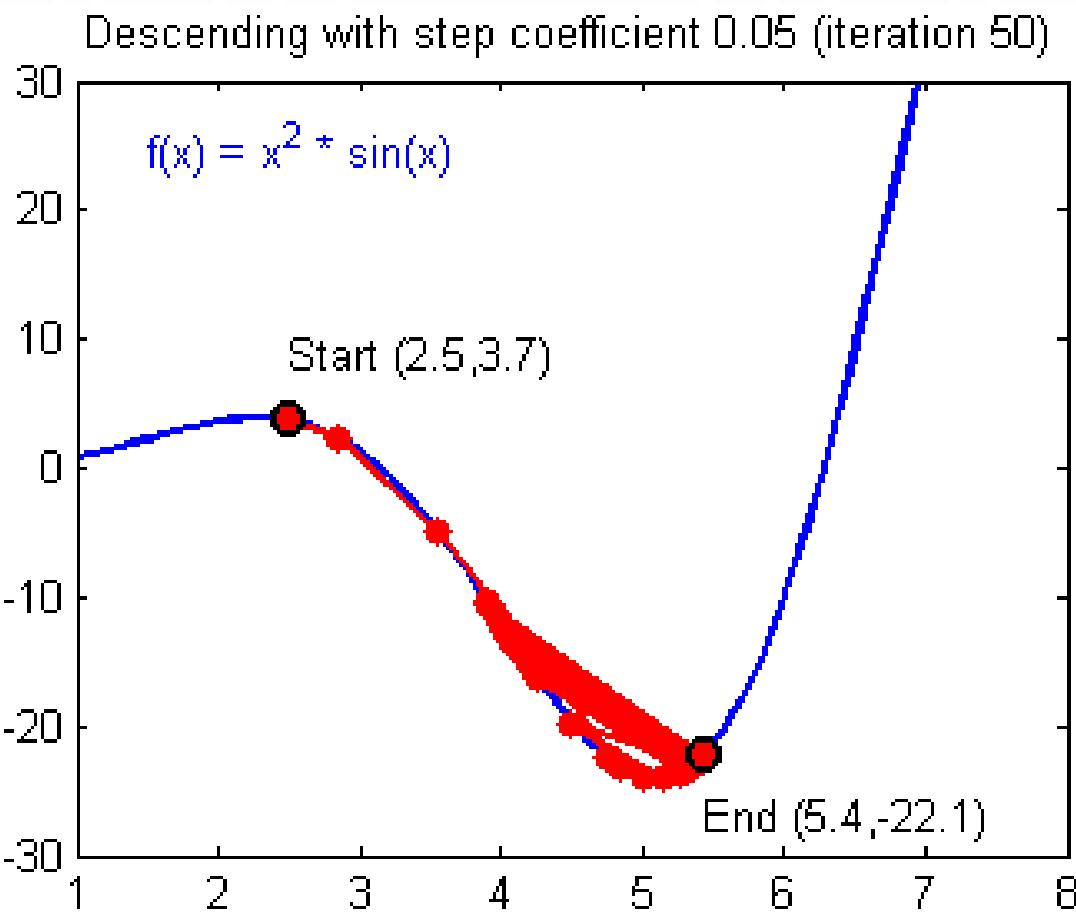
# КРИТЕРИИ ОСТАНОВА

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

# ГРАДИЕНТНЫЙ СПУСК



# ПРОБЛЕМА ВЫБОРА ГРАДИЕНТНОГО ШАГА



# 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

# МЕТОДЫ ОЦЕНИВАНИЯ ГРАДИЕНТА: SGD

## 1) Stochastic gradient descent (SGD):

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

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

# МЕТОДЫ ОЦЕНИВАНИЯ ГРАДИЕНТА: SGD

## 1) Stochastic gradient descent (SGD):

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

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

Скорость сходимости:  $E[Q(w^{(k)}) - Q(w^*)] = O(\frac{1}{\sqrt{k}})$

# МЕТОДЫ ОЦЕНИВАНИЯ ГРАДИЕНТА: SGD

## 1) Stochastic gradient descent (SGD):

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

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

Скорость сходимости:  $E[Q(w^{(k)}) - Q(w^*)] = O(\frac{1}{\sqrt{k}})$

+ Менее трудоемкий метод

- Медленнее сходится

# МЕТОДЫ ОЦЕНИВАНИЯ ГРАДИЕНТА: SAG

## 2) Stochastic average gradient (SAG):

- Инициализируем веса  $w_j$
- Инициализируем **вспомогательные переменные**  
 $z^{(1)}, z^{(2)}, \dots$ :

$$z^{(i)} = \nabla q_i(w)$$

# МЕТОДЫ ОЦЕНИВАНИЯ ГРАДИЕНТА: SAG

## 2) Stochastic average gradient (SAG):

- Инициализируем веса  $w_j$
- Инициализируем вспомогательные переменные  $z^{(1)}, z^{(2)}, \dots$ :

$$z^{(i)} = \nabla q_i(w)$$

- На каждом шаге выбираем **один случайный объект** и обновляем градиент по нему (все остальные градиенты остаются такими же, как на предыдущем шаге):

$$z_i^{(k)} = \begin{cases} \nabla q_i(w^{(k-1)}), & i = i_k \\ z_i^{(k-1)}, & \text{иначе} \end{cases}$$

# МЕТОДЫ ОЦЕНИВАНИЯ ГРАДИЕНТА: SAG

## 2) Stochastic average gradient (SAG):

- Инициализируем веса  $w_j$
- Инициализируем вспомогательные переменные  $z^{(1)}, z^{(2)}, \dots$ :

$$z^{(i)} = \nabla q_i(w)$$

- На каждом шаге выбираем один случайный объект и обновляем градиент по нему:

$$z_i^{(k)} = \begin{cases} \nabla q_i(w^{(k-1)}), & i = i_k \\ z_i^{(k-1)}, & \text{иначе} \end{cases}$$

- Формула градиентного шага:

$$w^{(k)} = w^{(k-1)} - \eta_k \sum_{i=1}^l z_i^{(k)}$$

# МЕТОДЫ ОЦЕНИВАНИЯ ГРАДИЕНТА: SAG

2) Stochastic average gradient (SAG):

- Формула градиентного шага:

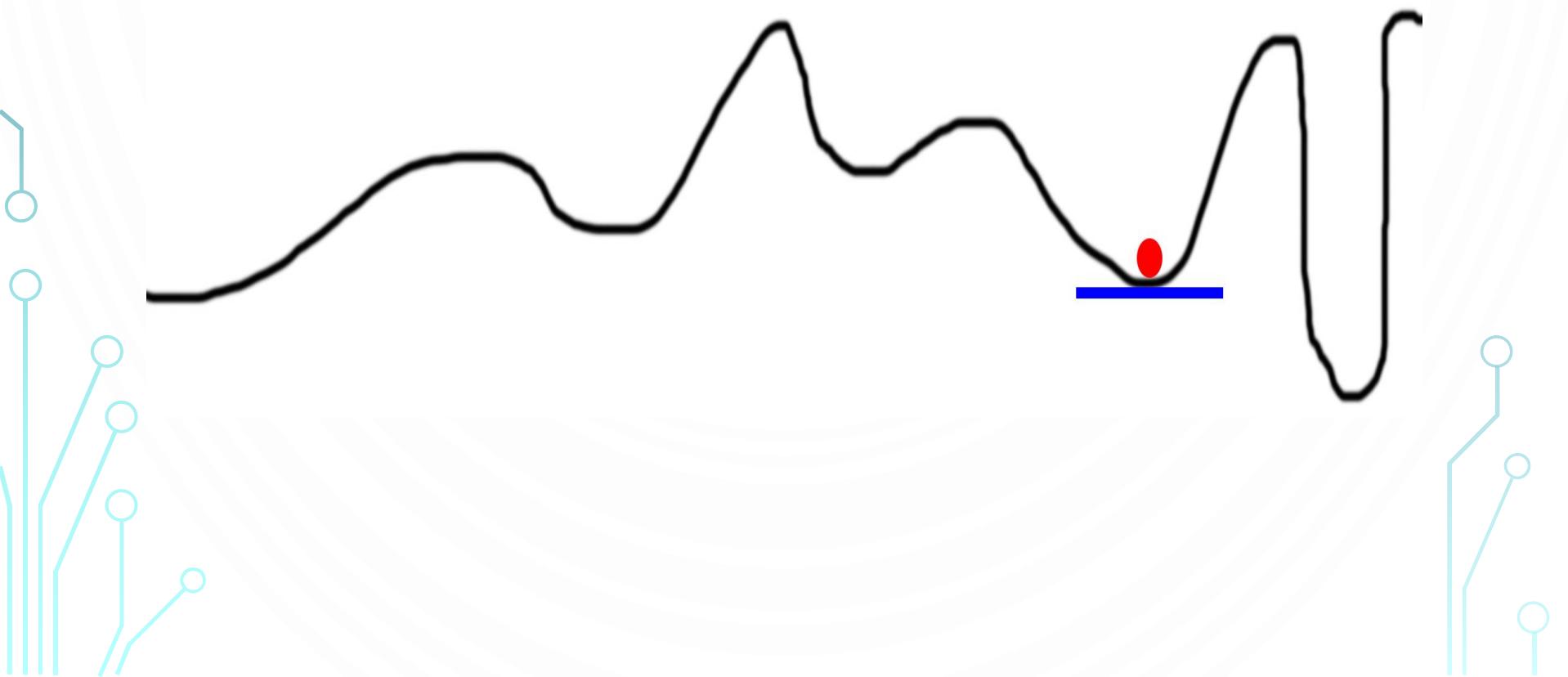
$$w^{(k)} = w^{(k-1)} - \eta_k \sum_{i=1}^l z_i^{(k)}$$

Скорость сходимости:  $E[Q(w^{(k)}) - Q(w^*)] = O(\frac{1}{k})$

# ПРОБЛЕМЫ ГРАДИЕНТНОГО СПУСКА

- Медленно сходится
- Застревает в локальных минимумах

# ПРОБЛЕМА ЗАСТРЕВАНИЯ В LOCMIN



# МЕТОД МОМЕНТОВ (МОМЕНТУМ)

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

$$h_0 = 0$$

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

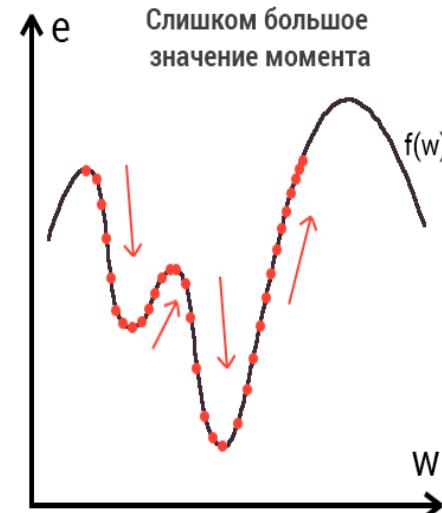
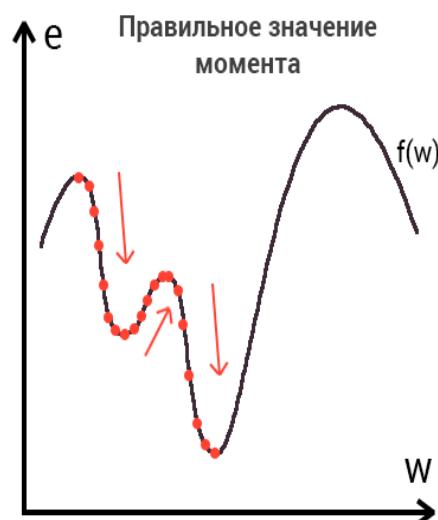
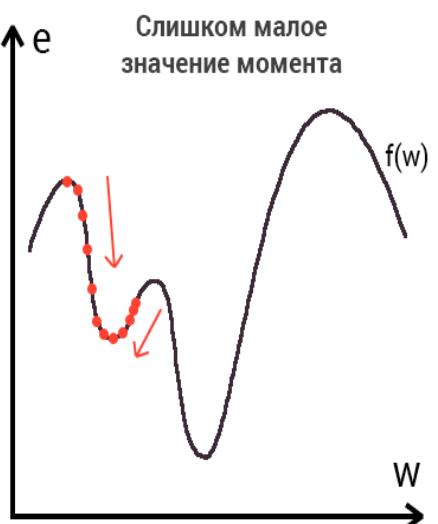
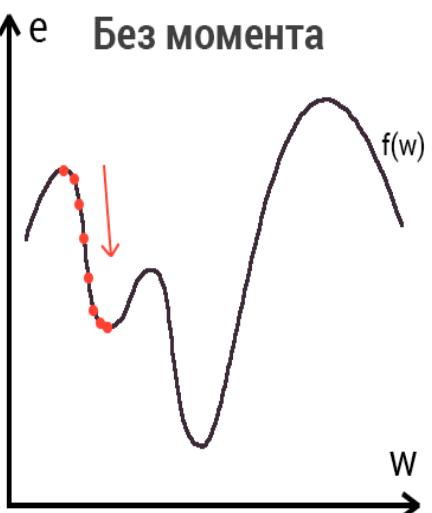
Формула метода моментов:

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

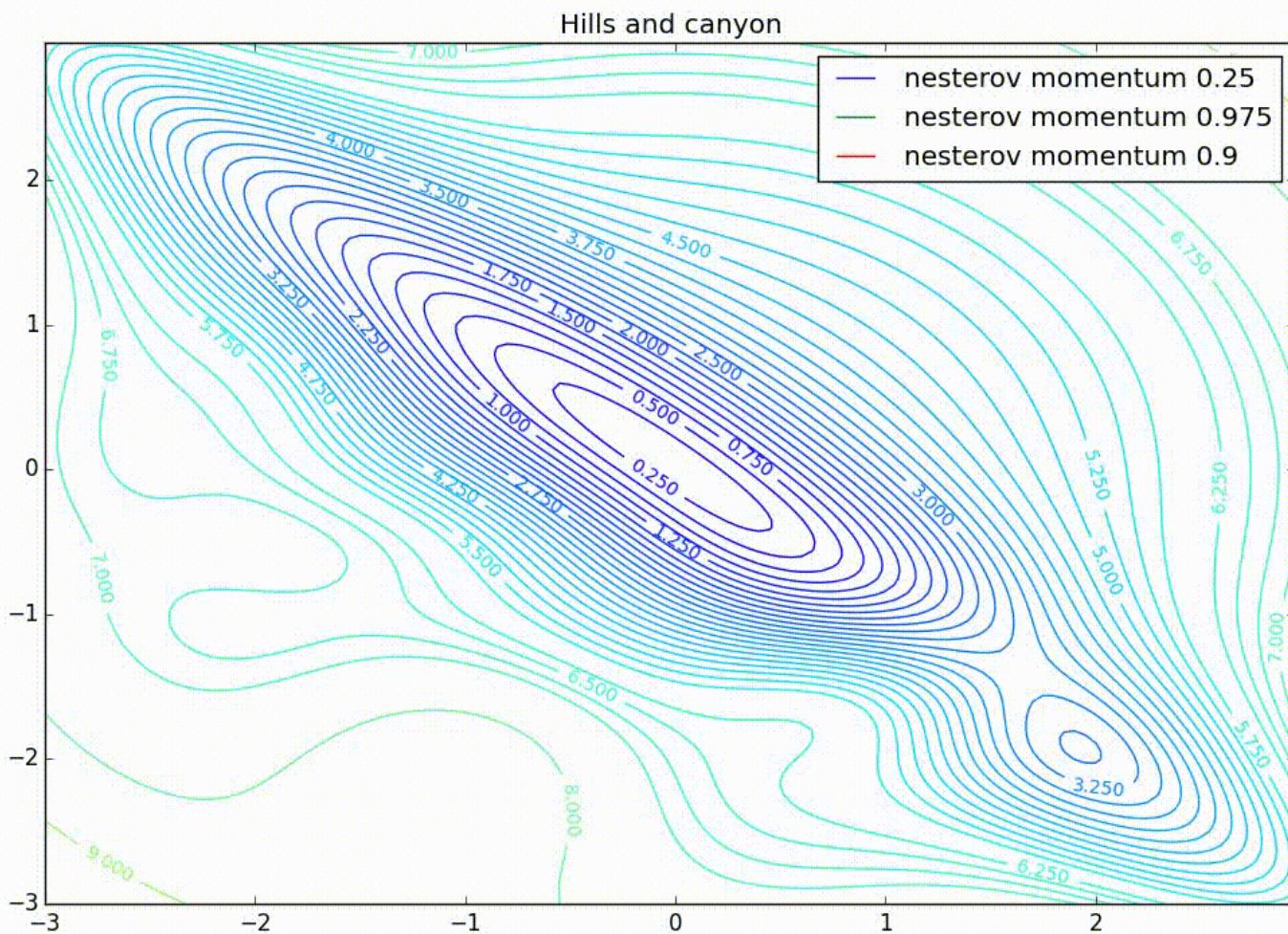
Подробнее:

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

# 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

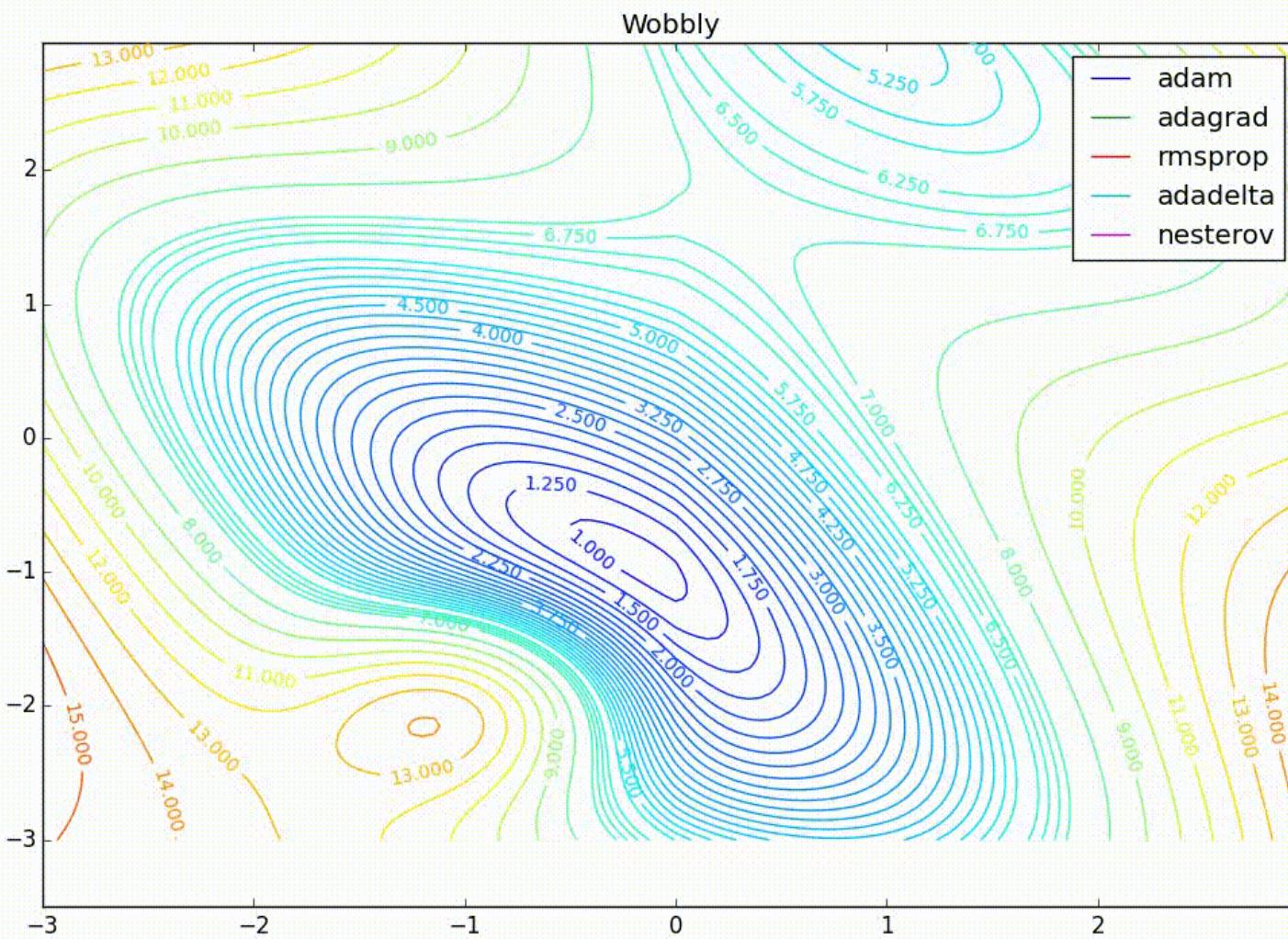
# RMSPROP (ROOT MEAN SQUARE PROPAGATION)

*Метод реализует экспоненциальное затухание градиентов*

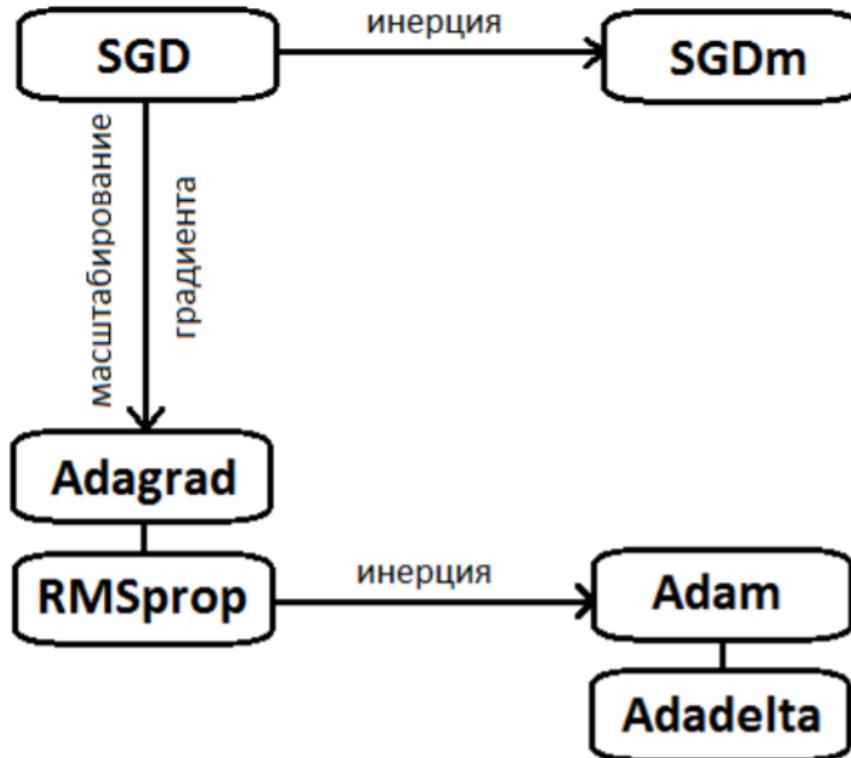
Формулы метода RMSprop (*усредненный по истории квадрат градиента*):

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

# МОДИФИКАЦИИ ГРАДИЕНТНОГО СПУСКА



# МОДИФИКАЦИИ SGD



[ссылка на статью со схемой](#)