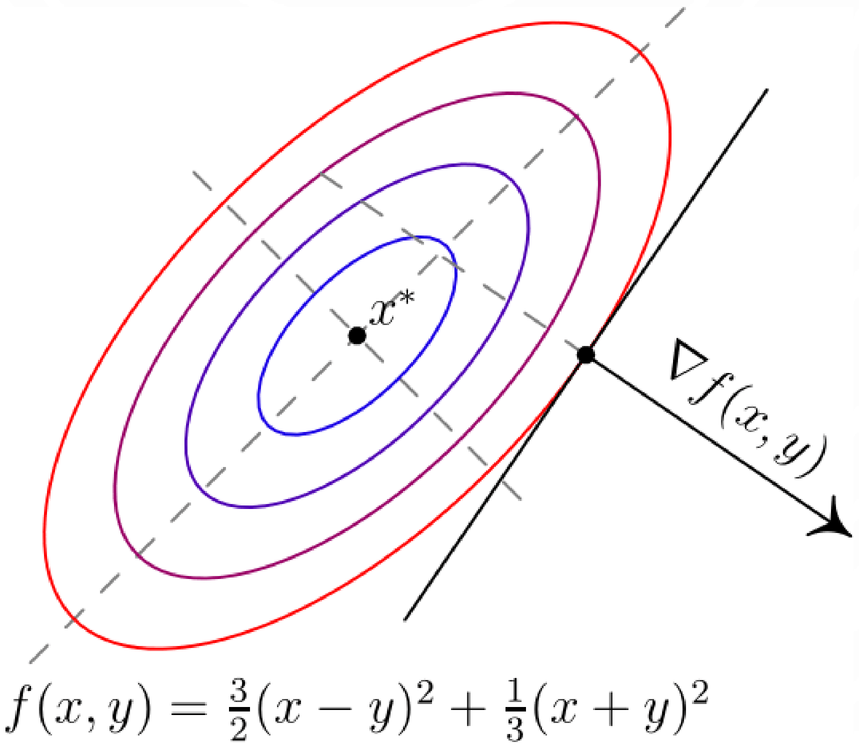
ТЕОРЕМА О ГРАДИЕНТЕ

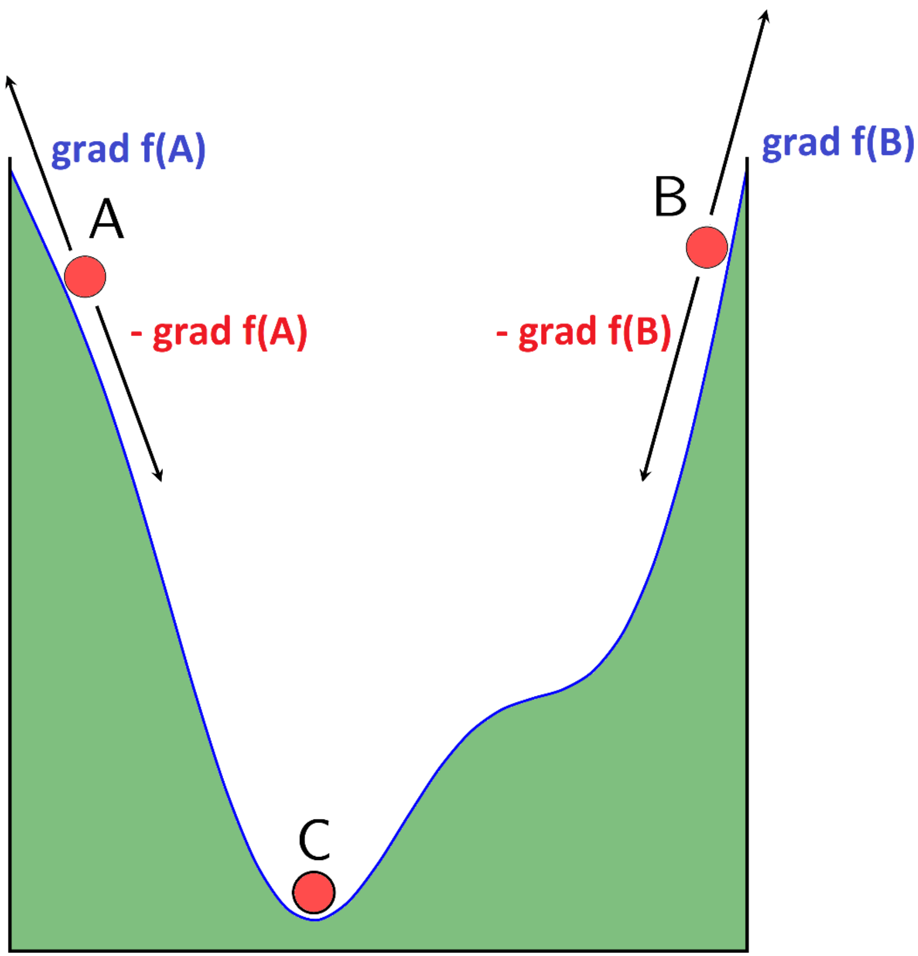
Теорема. Градиент – это вектор, в направлении которого функция быстрее всего растёт.

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



ТЕОРЕМА О ГРАДИЕНТЕ

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



МЕТОД ГРАДИЕНТНОГО СПУСКА

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

Метод градиентного спуска

- Наша задача при обучении модели – найти такие веса $\boldsymbol{w}$, 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 $\boldsymbol{w}$, 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 потери!

То есть на каждом шаге движемся в направлении уменьшения ошибки.

МЕТОД ГРАДИЕНТНОГО СПУСКА

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

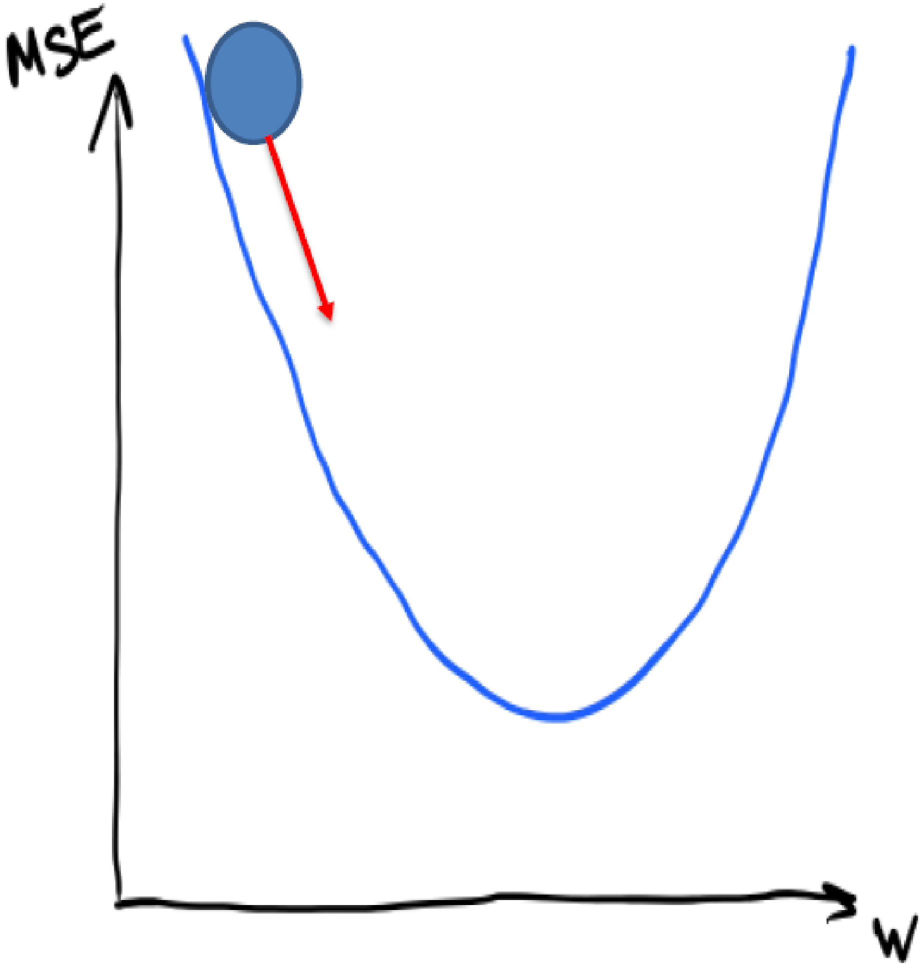
На каждом шаге (на каждой итерации метода) движемся в сторону антиградиента функции потерь!

То есть на каждом шаге движемся в направлении уменьшения ошибки.

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

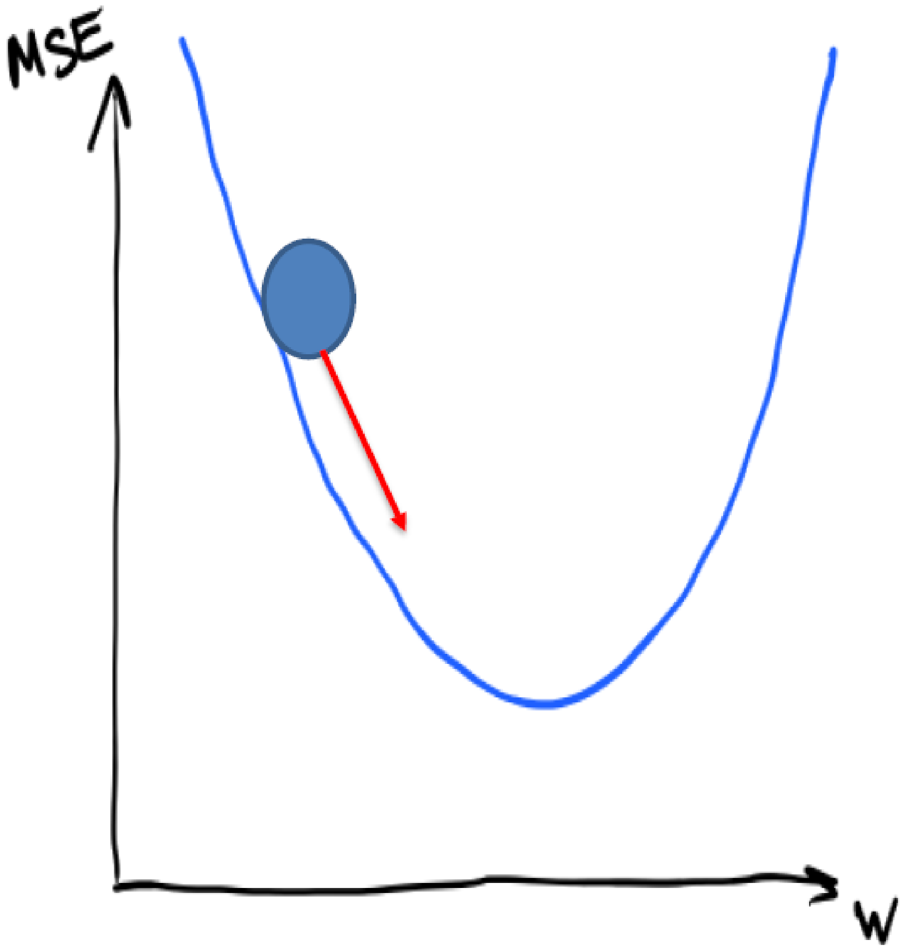
Метод градиентного спуска

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



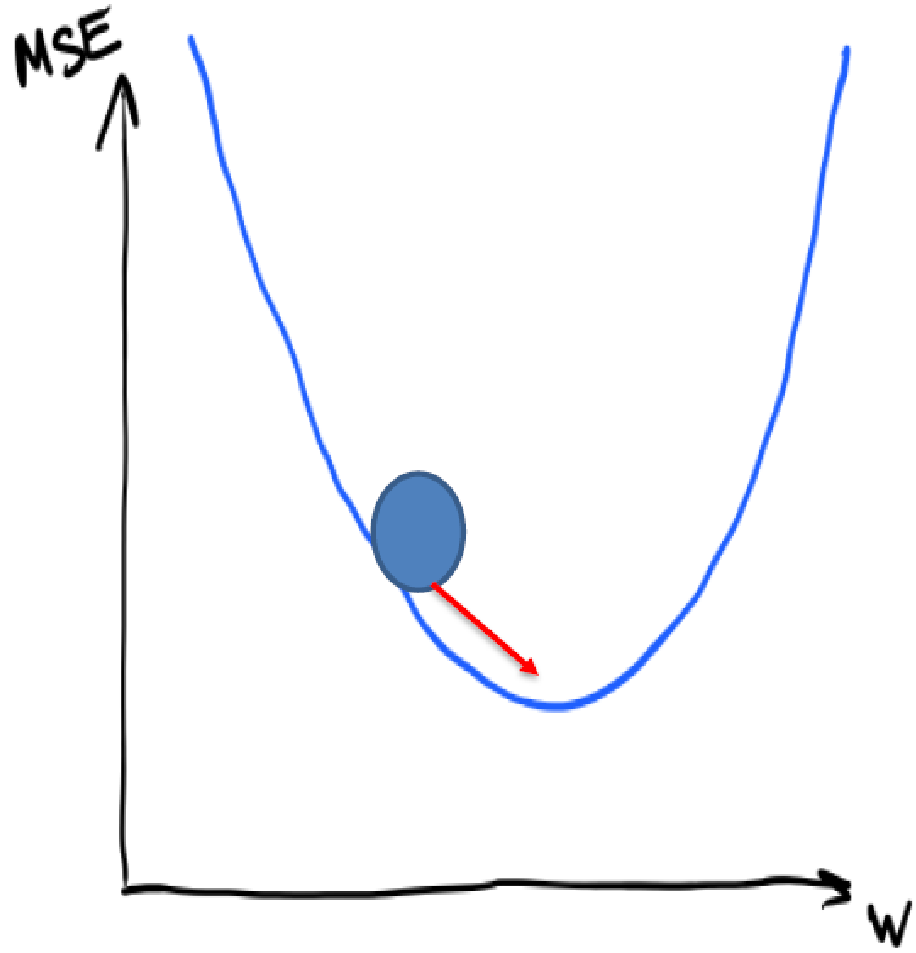
Метод градиентного спуска

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



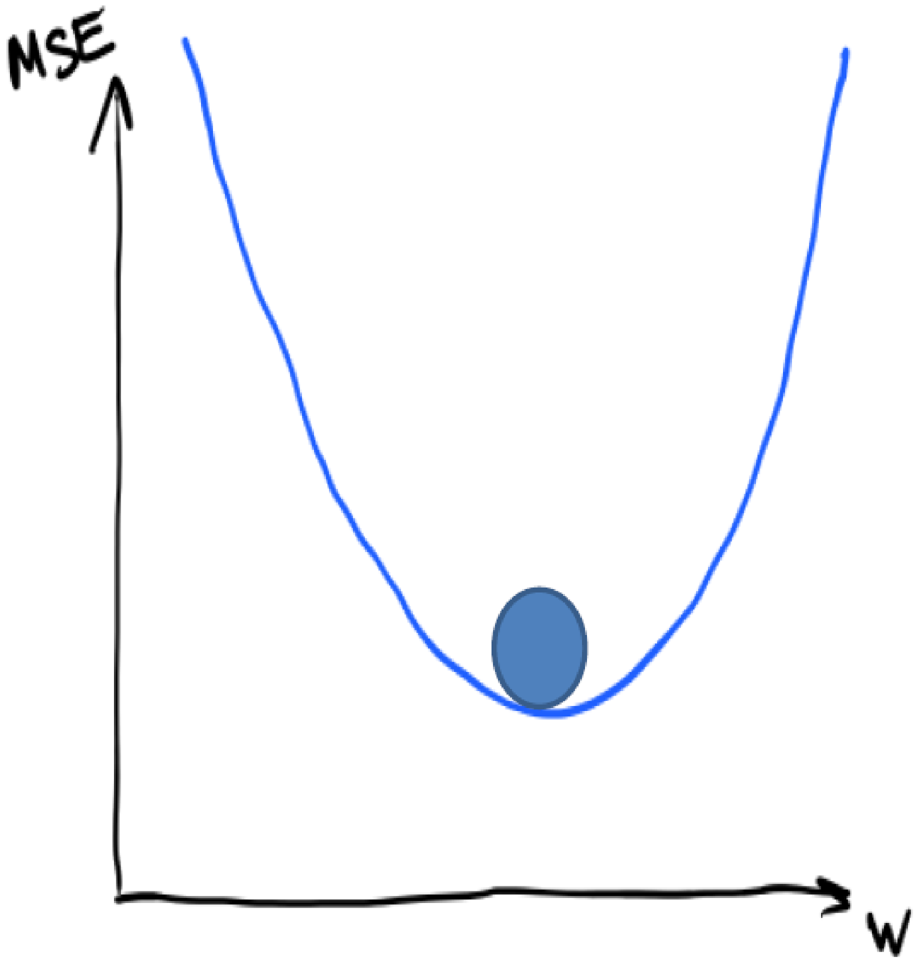
Метод градиентного спуска

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):

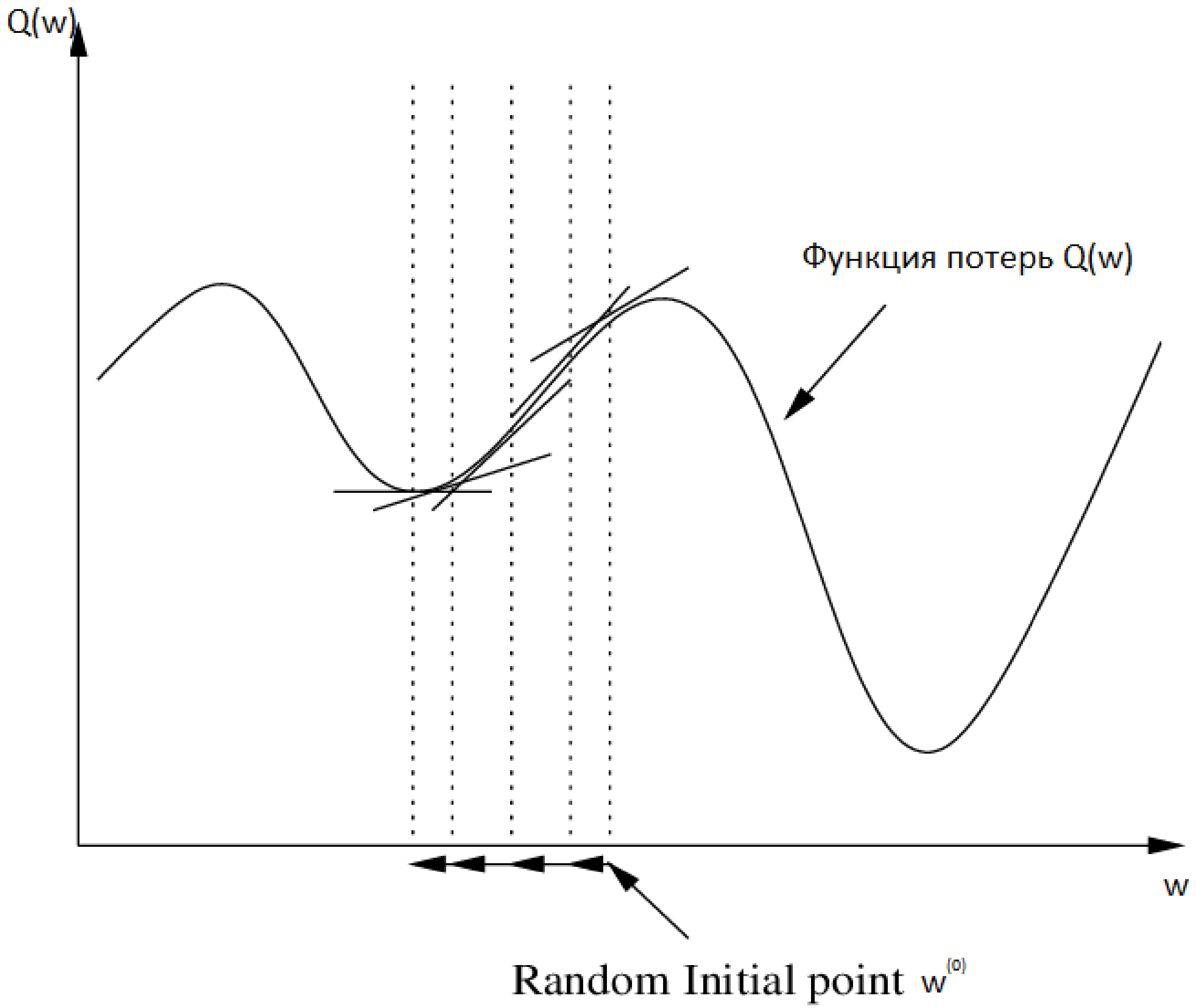
Пусть у нас только один вес - w.

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

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

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

МЕТОД ГРАДИЕНТНОГО СПУСКА

Метод градиентного спуска (общий случай случай):

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

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

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

Метод градиентного спуска

Метод градиентного спуска (общий случай случай):

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

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

- Инициализируем веса $w_0^{(0)}, w_1^{(0)}, w_2^{(0)}, \dots, w_n^{(0)}$.
- На каждом следующем шаге обновляем веса:

$$w_0^{(k)} = w_0^{(k-1)} - \frac{\partial Q}{\partial w_0}\left(w_0^{(k-1)}\right),$$

$$\dots$$

$$w_n^{(k)} = w_n^{(k-1)} - \frac{\partial Q}{\partial w_n}\left(w_n^{(k-1)}\right).$$

МЕТОД ГРАДИЕНТНОГО СПУСКА

Формулу для обновления весов можно записать в векторном виде:

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

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

МЕТОД ГРАДИЕНТНОГО СПУСКА

Формулу для обновления весов можно записать в векторном виде:

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

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

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

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

МЕТОД ГРАДИЕНТНОГО СПУСКА

Формулу для обновления весов можно записать в векторном виде:

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

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

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

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

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^*.

МЕТОД ГРАДИЕНТНОГО СПУСКА

Пример (решение на доске):

выписать формулы обновления весов методом градиентного спуска.

$$y = w_0 + w_1 x$$

$$Q(w) = \sum_{i=1}^{l} (w_0 + w_1 x_i - y_i)^2$$

ВАРИАНТЫ ИНИЦИАЛИЗАЦИИ ВЕСОВ

- $w_j = 0, j = 1, \ldots, n$
- Небольшие случайные значения:

$$w_j := random(-\varepsilon, \varepsilon)$$

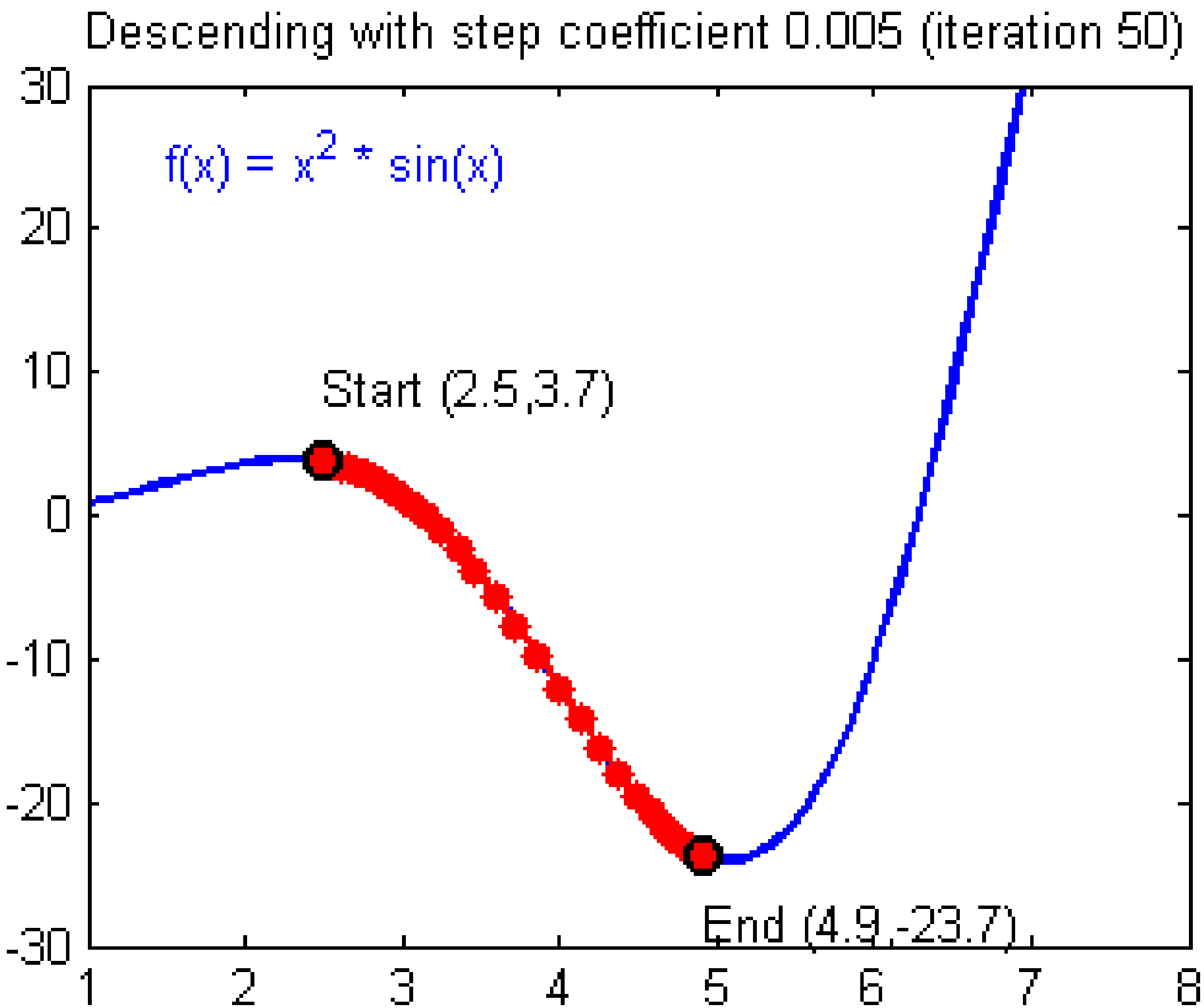
- Обучение по небольшой случайной подвыборке объектов
- Мультистарт: многократный запуск из разных случайных начальных приближений и выбор лучшего решения

КРИТЕРИИ ОСТАНОВА

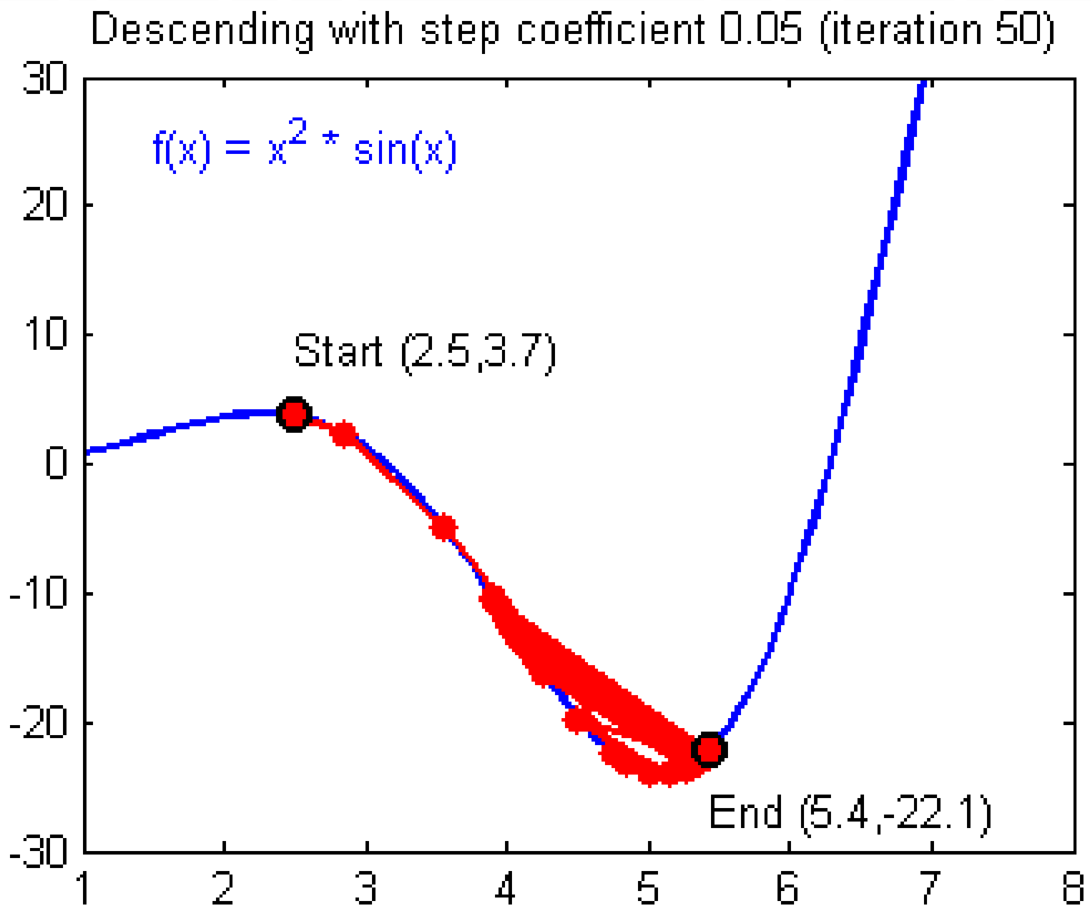
- $\left|Q\left(w^{(k)}\right) - Q\left(w^{(k-1)}\right)\right| < \varepsilon$

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

ГРАДИЕНТНЫЙ СПУСК



ПРОБЛЕМА ВЫБОРА ГРАДИЕНТНОГО ШАГА



Градиентный шаг

В общем случае градиентный шаг может зависеть от номера итерации, тогда будем писать не 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

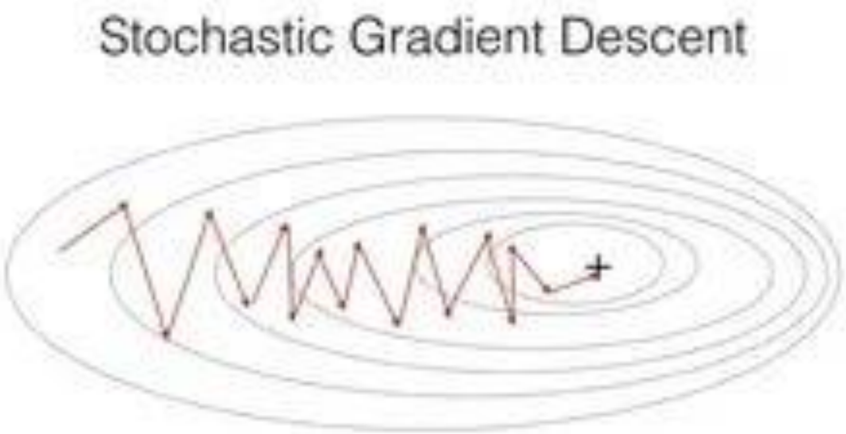
Stochastic gradient descent (SGD):

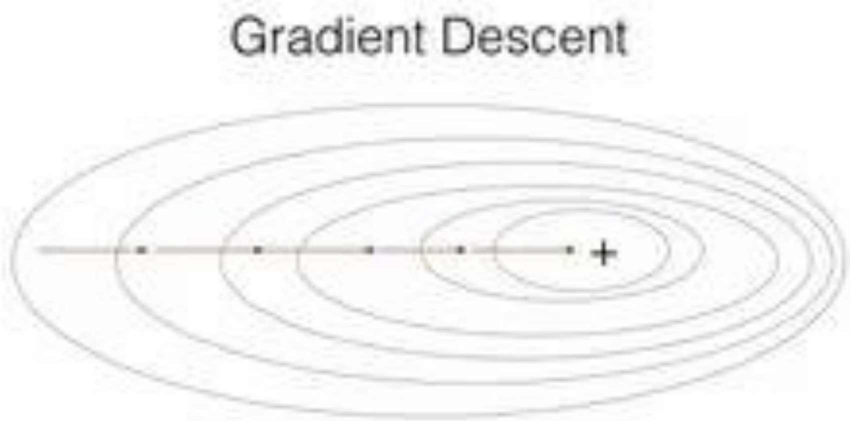
- на каждом шаге выбираем ***один случайный объект*** и сдвигаемся в сторону антиградиента по этому объекту:

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

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

СТОХАСТИЧЕСКИЙ ГРАДИЕНТНЫЙ СПУСК





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

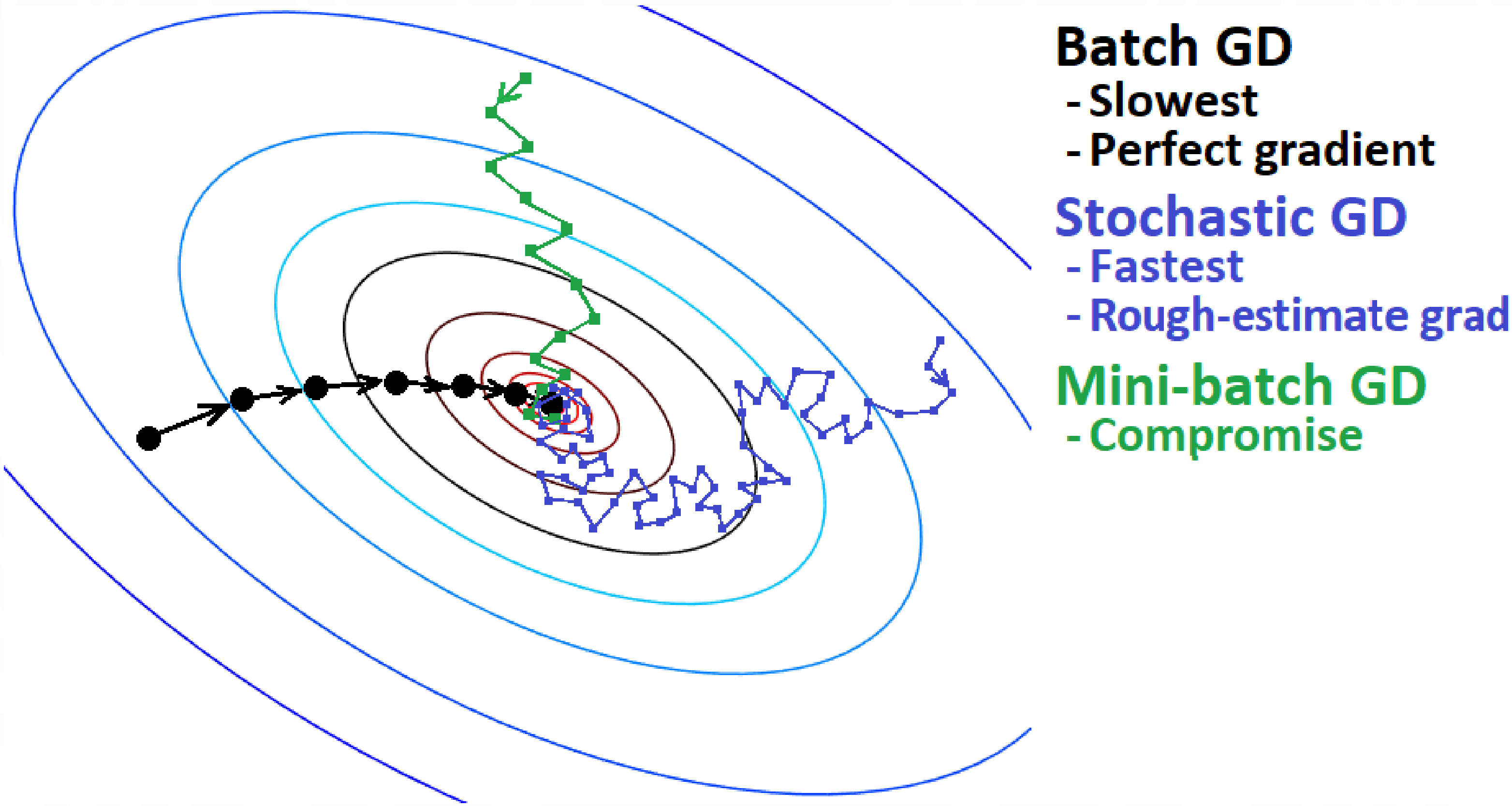
Промежуточное решение между классическим градиентным спуском и стохастическим вариантом.

- 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:

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

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

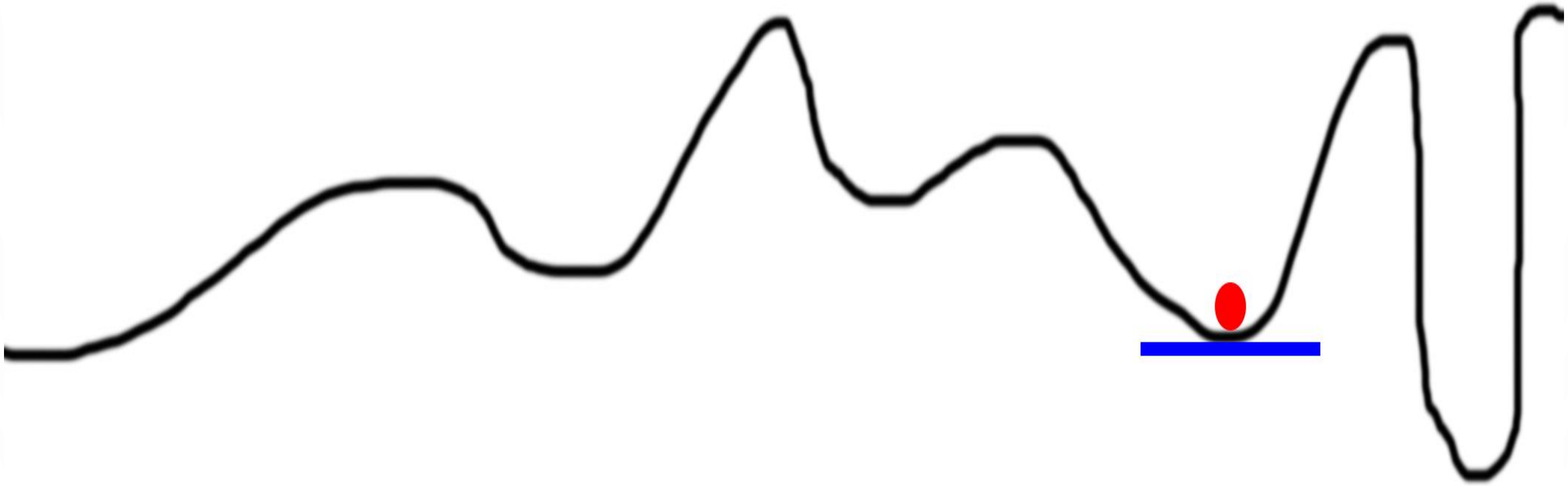
ВАРИАНТЫ ГРАДИЕНТНОГО СПУСКА



БОНУС: ПРОБЛЕМЫ ГРАДИЕНТНОГО СПУСКА И ВАРИАНТЫ ИХ РЕШЕНИЯ

- Медленно сходится
- Застревает в локальных минимумах

ПРОБЛЕМА ЗАСТРЕВАНИЯ В LOCMIN



Метод моментов (Momentum)

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

$$h_0 = 0$$

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

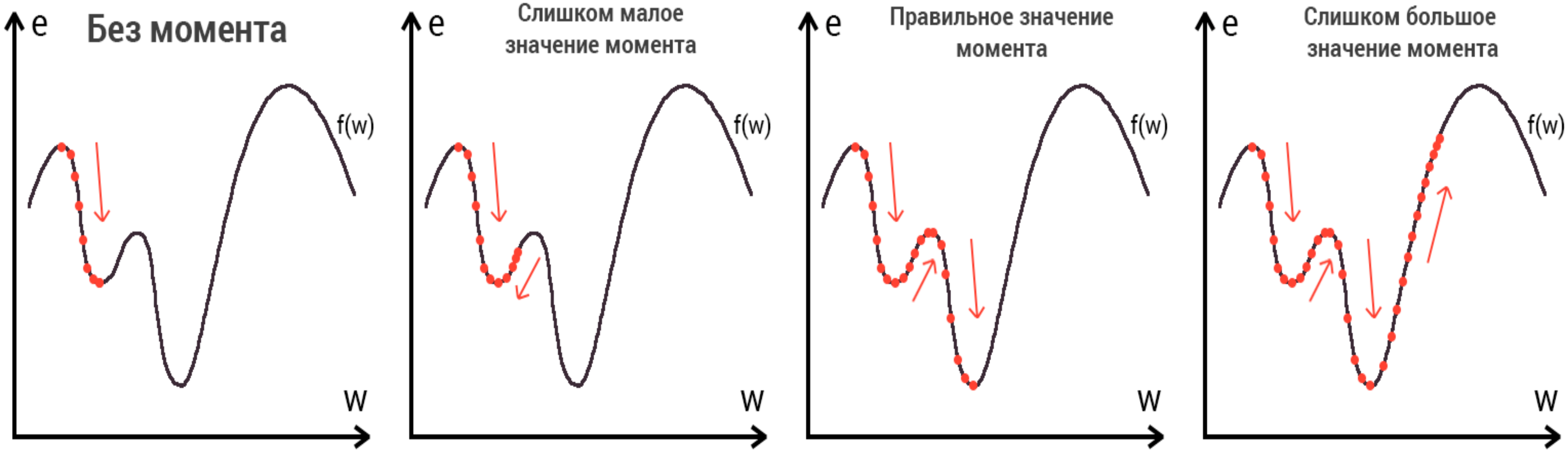
Формула метода моментов:

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

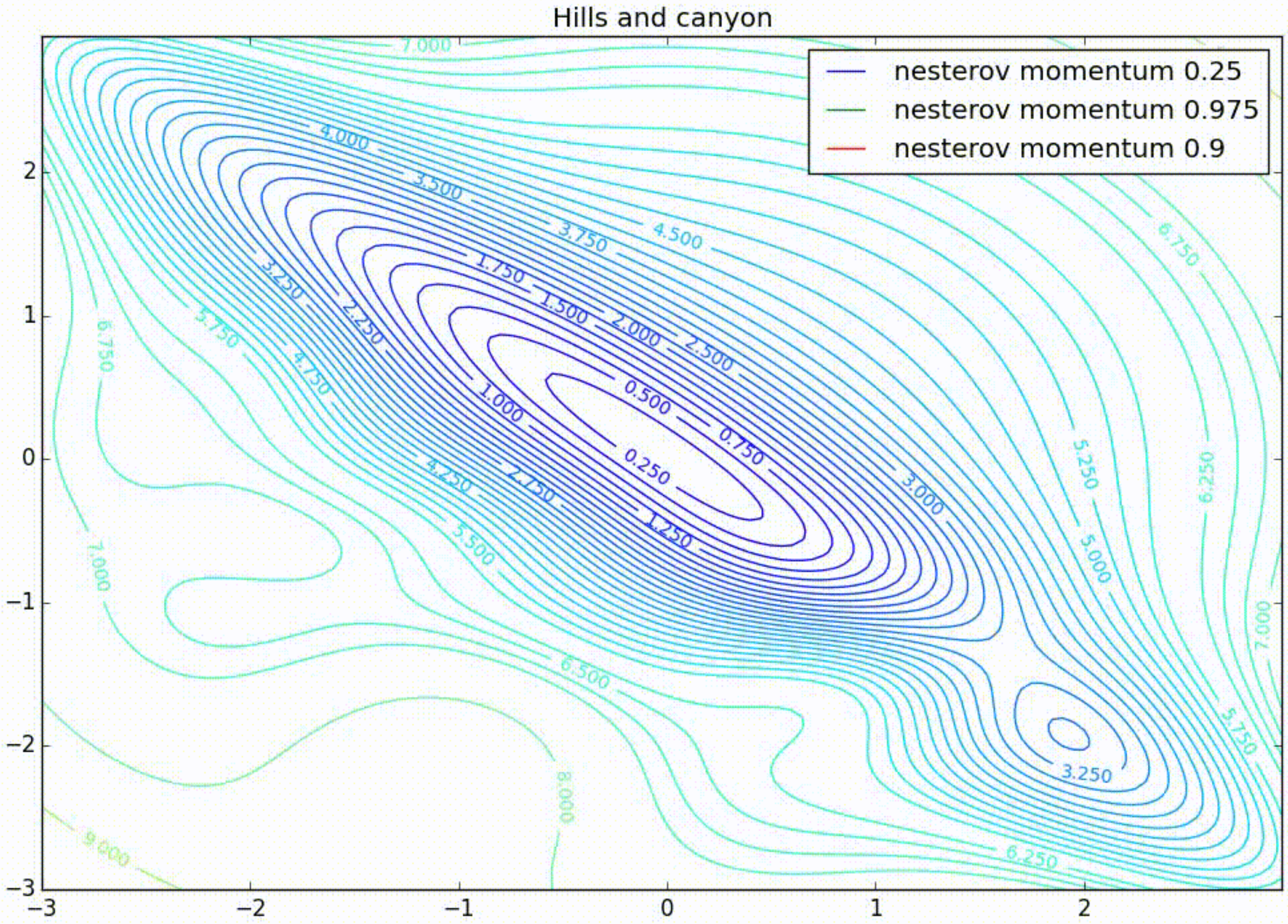
Подробнее:

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

MOMENTUM


e
Без момента
f(w)
w
e
Слишком малое значение момента
f(w)
w
e
Правильное значение момента
f(w)
w
e
Слишком большое значение момента
f(w)
w

MOMENTUM



Hills and canyon
nesterov momentum 0.25
nesterov momentum 0.975
nesterov momentum 0.9

ADAGRAD (ADAPTIVE GRADIENT)

Сумма квадратов обновлений:

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

Формулы метода AdaGrad:

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

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:

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

+ Автоматическое затухание скорости обучения

- G_{kj} монотонно возрастают, поэтому шаги укорачиваются, и мы можем не успеть дойти до минимума

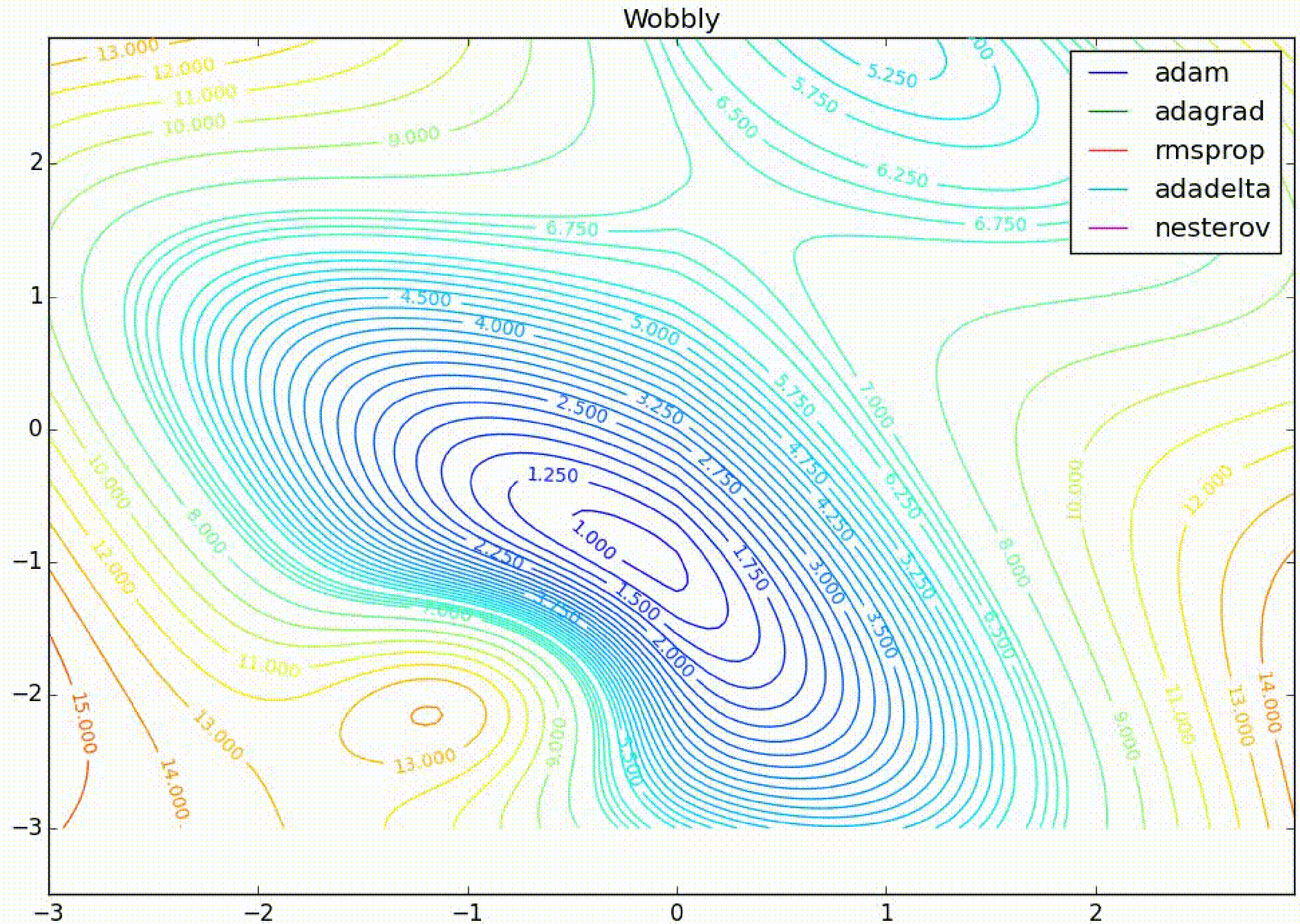
RMSPROP (ROOT MEAN SQUARE PROPAGATION)

Метод реализует экспоненциальное затухание градиентов

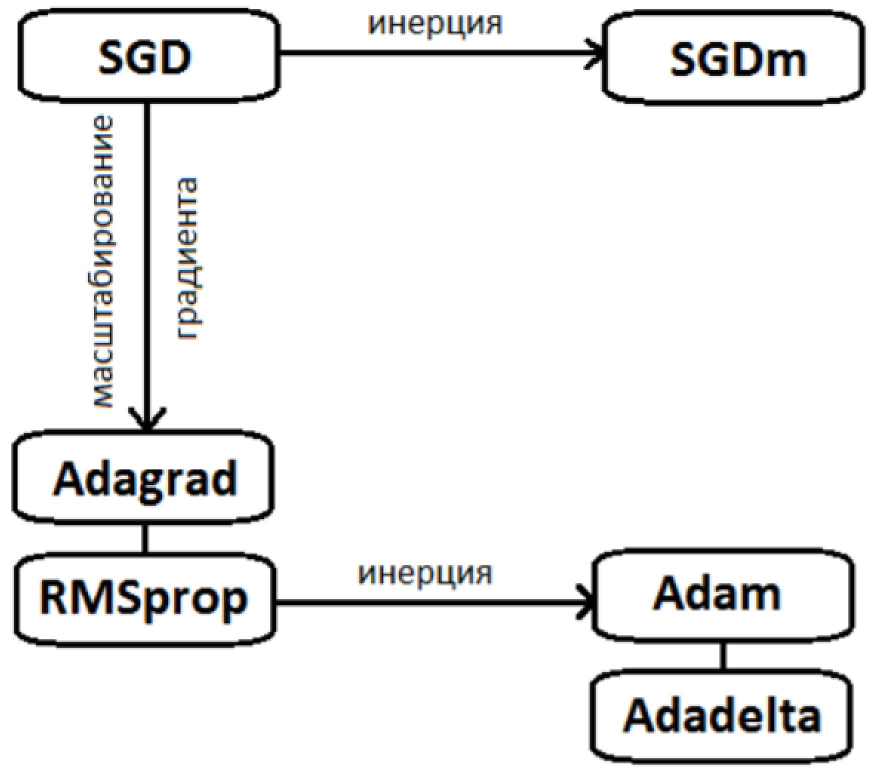
Формулы метода RMSprop (*усредненный по истории квадрат градиента*):

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

МОДИФИКАЦИИ ГРАДИЕНТНОГО СПУСКА



МОДИФИКАЦИИ SGD



[ссылка на статью]