

# Машинное обучение

Лекция 3

Градиентные методы обучения

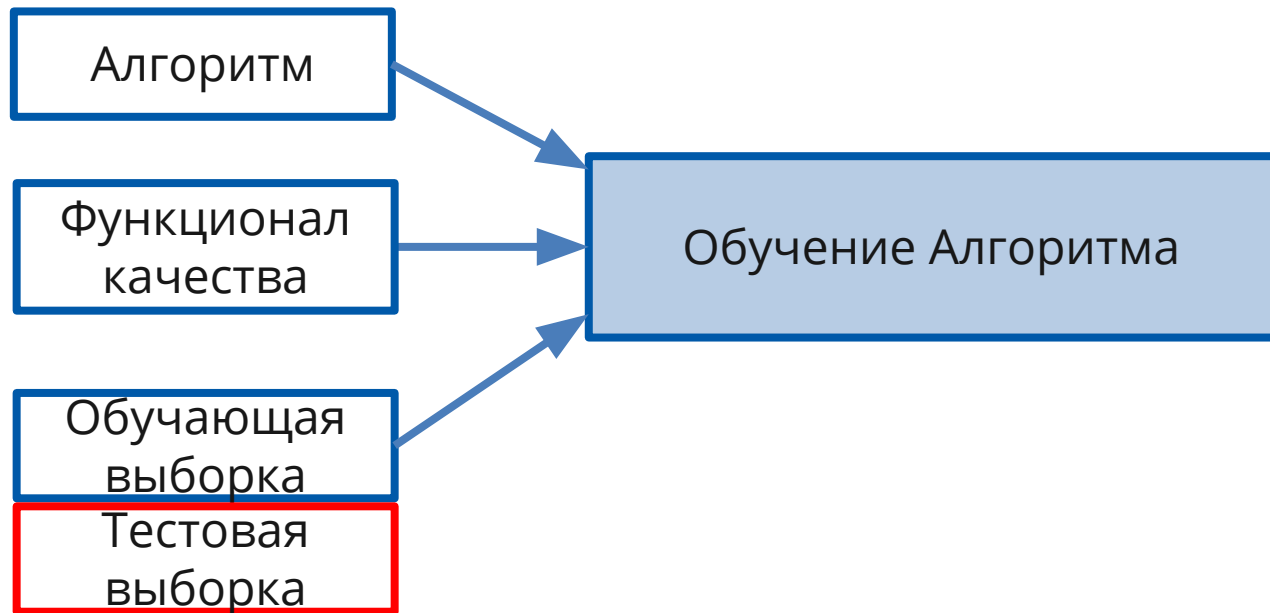
Аня Кузина

[https://github.com/AKuzina/ml\\_dpo](https://github.com/AKuzina/ml_dpo)

# План

- Резюме прошлой лекции
  - Линейная регрессия
  - Решение в явном виде и его проблемы
  - Регуляризация
- Градиентный спуск для решения задач оптимизации
- Модификации градиентного спуска

# Обучение алгоритма



# Линейная Регрессия

Алгоритм

Функционал  
качества

Обучающая  
выборка

Тестовая  
выборка

# Линейная Регрессия

Алгоритм

$$a(x) = w_1x_1 + w_2x_2 + \dots + w_dx_d$$

Функционал  
качества

$$Q(w) = \frac{1}{l} \|Xw - y\|^2$$

Обучающая  
выборка

Тестовая  
выборка

$$X = \begin{bmatrix} x_{11} & \dots & x_{1d} \\ \vdots & & \vdots \\ x_{l1} & \dots & x_{ld} \end{bmatrix} \in \mathbb{R}^{l \times d} \quad y = \begin{bmatrix} y_1 \\ \vdots \\ y_l \end{bmatrix} \in \mathbb{R}^l$$

# Обучение Линейной Регрессии

$$Q(w) = \frac{1}{l} \|Xw - y\|^2 \rightarrow \min_w$$

1. Аналитическое решение
2. Итерационные методы оптимизации

# Обучение Линейной Регрессии

$$Q(w) = \frac{1}{l} \|Xw - y\|^2 \rightarrow \min_w$$

## 1. Аналитическое решение

# Обучение Линейной Регрессии

$$Q(w) = \frac{1}{l} \|Xw - y\|^2 \rightarrow \min_w$$

## 1. Аналитическое решение

- Градиент

$$\nabla Q(w) = \frac{2}{l} (X^T X w - X^T y)$$

- Решение

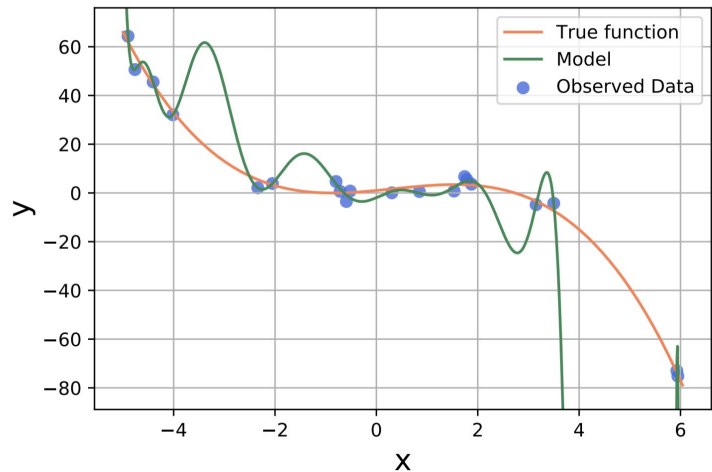
$$\nabla Q(w) = 0$$

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



# Проблемы

## Переобучение



## Аналитическое решение на практике

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

# Регуляризация

Задача обучения линейной регрессии

$$Q(w) = \frac{1}{l} \|Xw - y\|^2 \rightarrow \min_w$$

Добавим “штраф”:

$$Q(w) + \lambda R(w) \rightarrow \min_w$$

# Регуляризация

$$Q(w) + \lambda R(w) \rightarrow \min_w$$

Норма вектора в качестве  
регуляризатора

$$R(w) = \|w\|_2^2 = \sum_{j=1}^d w_j^2$$

$$R(w) = \|w\|_1 = \sum_{j=1}^d |w_j|$$

# Регуляризация

$$Q(w) + \lambda R(w) \rightarrow \min_w$$

Норма вектора в качестве  
регуляризатора

$$R(w) = \|w\|_2^2 = \sum_{j=1}^d w_j^2$$

-> Ridge regression

$$R(w) = \|w\|_1 = \sum_{j=1}^d |w_j|$$

-> Lasso regression

# Гребневая регрессия (Ridge)

$$\frac{1}{l} \|Xw - y\|^2 + \lambda \|w\|_2^2 \rightarrow \min_w$$

1. Аналитическое решение (более стабильное)

$$w^* = (X^T X + \lambda I)^{-1} X^T y$$

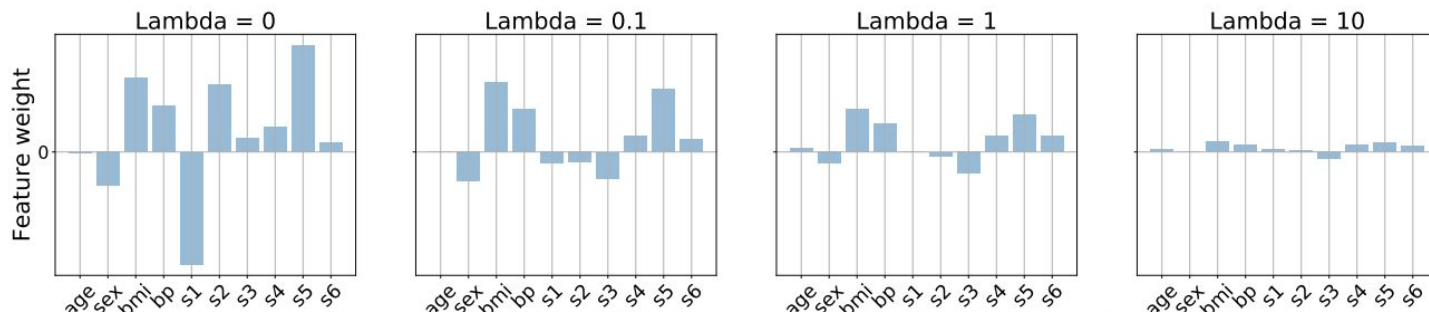
# Гребневая регрессия (Ridge)

$$\frac{1}{l} \|Xw - y\|^2 + \lambda \|w\|_2^2 \rightarrow \min_w$$

1. Аналитическое решение (более стабильное)

$$w^* = (X^T X + \lambda I)^{-1} X^T y$$

2. Эффект сокращения весов (shrinkage)



# Лассо регрессия (LASSO)

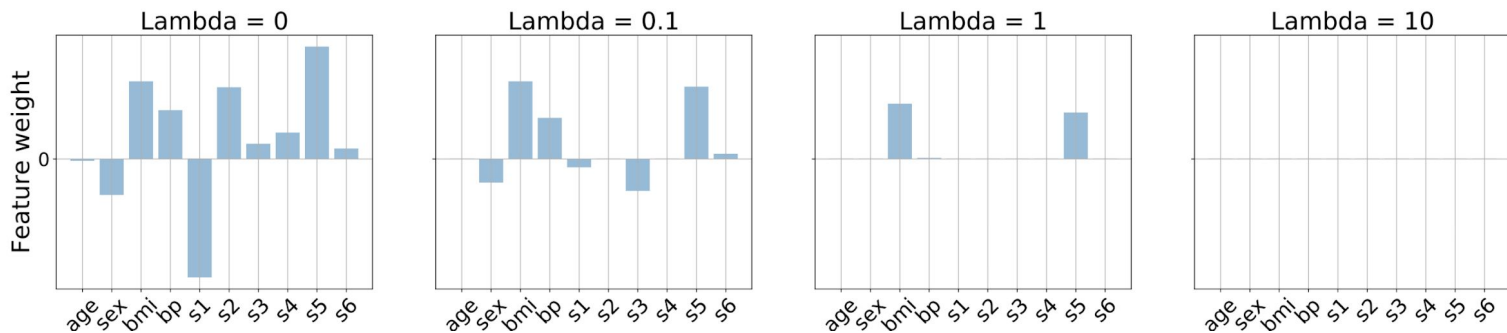
$$\frac{1}{l} \|Xw - y\|^2 + \lambda \|w\|_1 \rightarrow \min_w$$

1. Least Absolute Shrinkage and Selection Operator

# Лассо регрессия (LASSO)

$$\frac{1}{l} \|Xw - y\|^2 + \lambda \|w\|_1 \rightarrow \min_w$$

1. Least Absolute Shrinkage and Selection Operator
2. Эффект сокращения весов (shrinkage) и отбора признаков (selection)





Что если не аналитическое решение?

$$Q(w) + \lambda R(w) \rightarrow \min_w$$

Что если не аналитическое решение?

$$Q(w) + \lambda R(w) \rightarrow \min_w$$

Итерационные методы  
оптимизации

# Градиент

Вектор частных производных

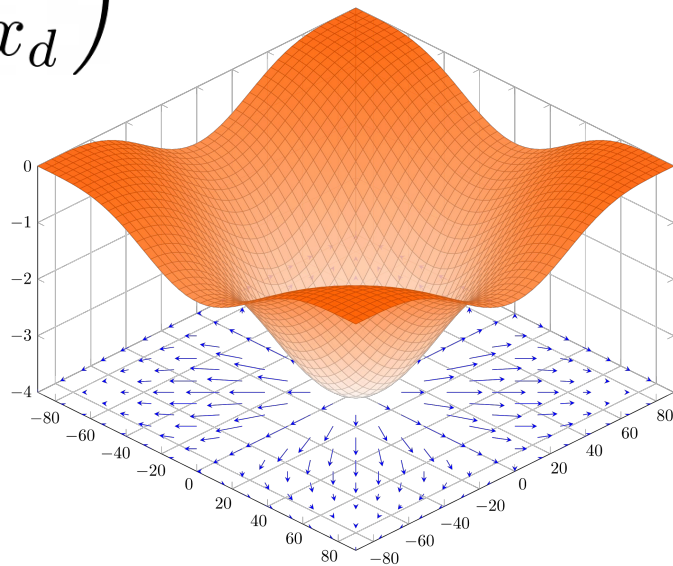
$$\nabla f(x) = \left( \frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_d} \right)$$

# Градиент

Вектор частных производных

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

Градиент показывает направление  
наискорейшего возрастания  
функции



## Градиент: условие экстремума

Если точка  $x_0$  — экстремум и в ней существует производная, то

$$\nabla f(x_0) = 0$$

# Градиентный спуск

Задача: найти минимум функции

$$\min_x f(x)$$

# Градиентный спуск



<https://www.codingame.com/playgrounds/9487/deep-learning-from-scratch---theory-and-implementation/gradient-descent-and-backpropagation>

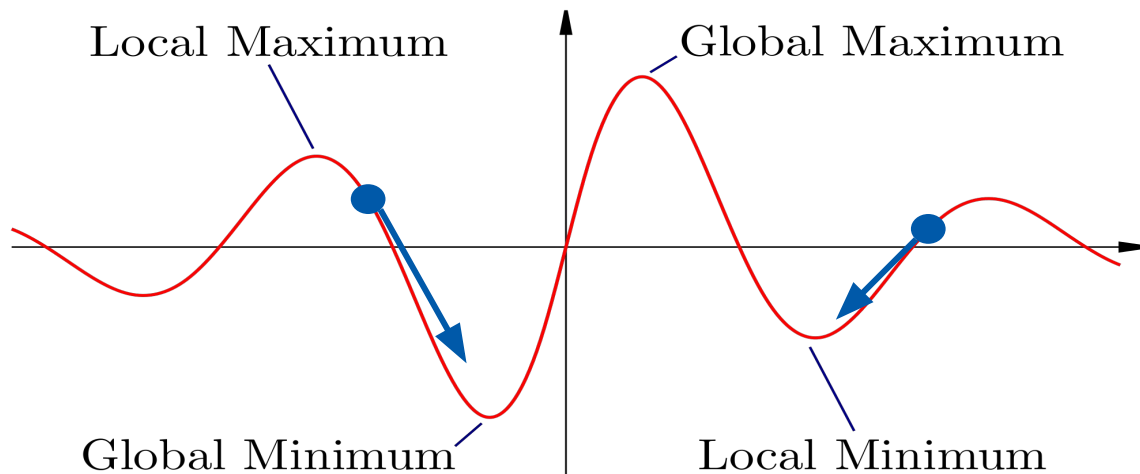
# Градиентный спуск

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



# Градиентный спуск: начальное приближение

- Стартуем из случайной точки
  - Как выбрать  $x_0$  ?



# Градиентный спуск: итерации

- Стартуем из случайной точки
- Сдвигаемся по антиградиенту

$$x^t = x^{t-1} - \eta \nabla f(x^{t-1})$$

Новая точка

Длина шага  
(Learning rate)


Градиент в  
предыдущей точке

# Градиентный спуск: сходимость

- Стартуем из случайной точки
- Сдвигаемся по антиградиенту
- Повторяем, пока не окажемся в точке минимума
  - $\|x^t - x^{t-1}\| < \varepsilon$
  - $\|f(x^t) - f(x^{t-1})\| < \varepsilon$
  - $\|\nabla f(x^t)\| < \varepsilon$

# Градиентный спуск: длина шага

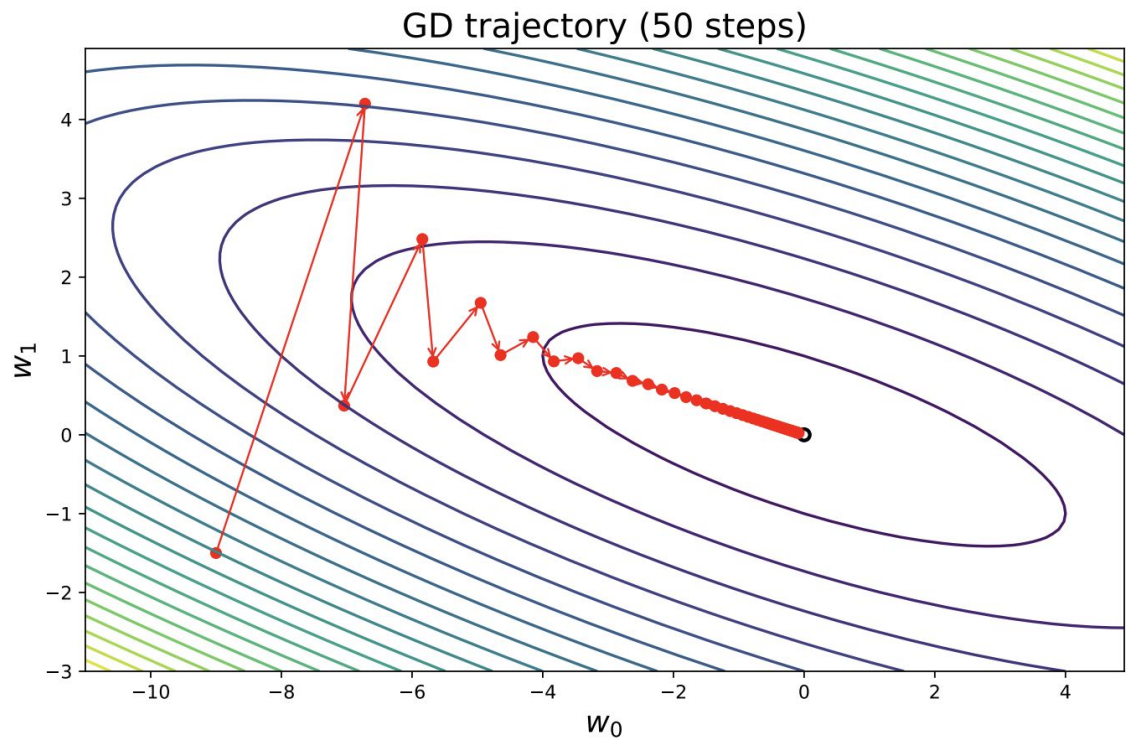
$$x^t = x^{t-1} - \eta \nabla f(x^{t-1})$$



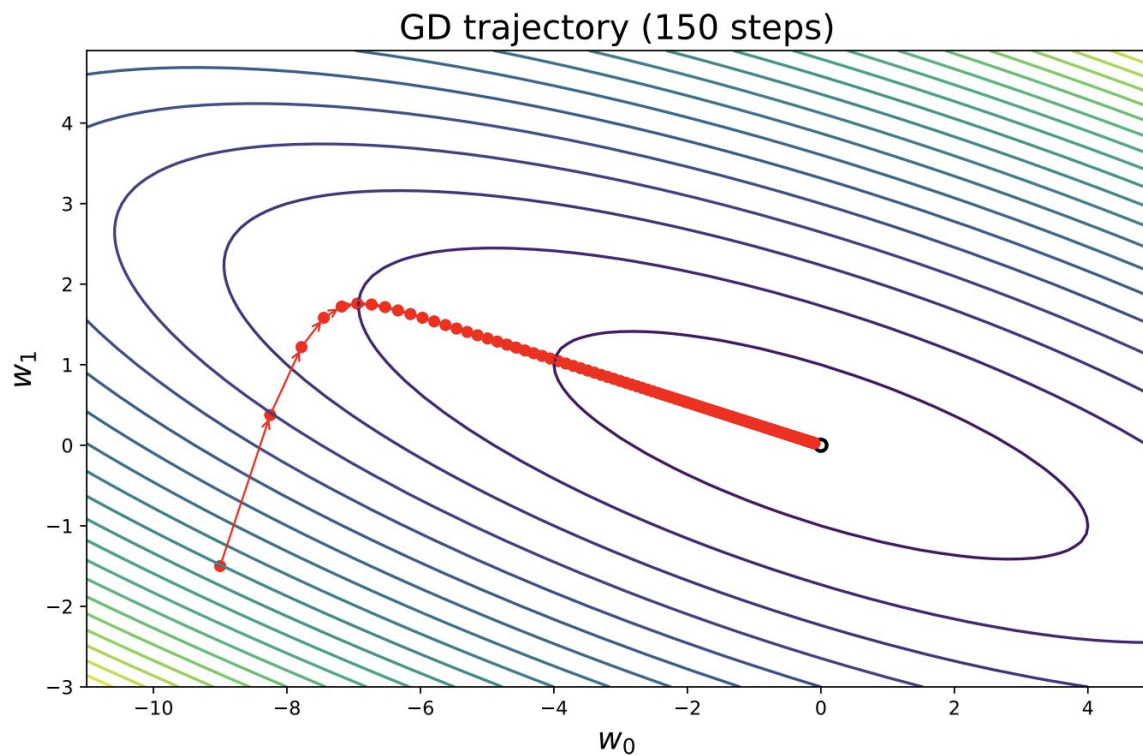
Длина шага  
(Learning rate)

- Позволяет контролировать скорость обучения
- Если сделать длину шага недостаточно маленькой, градиентный спуск может разойтись
- Длина шага — гиперпараметр, который нужно подбирать

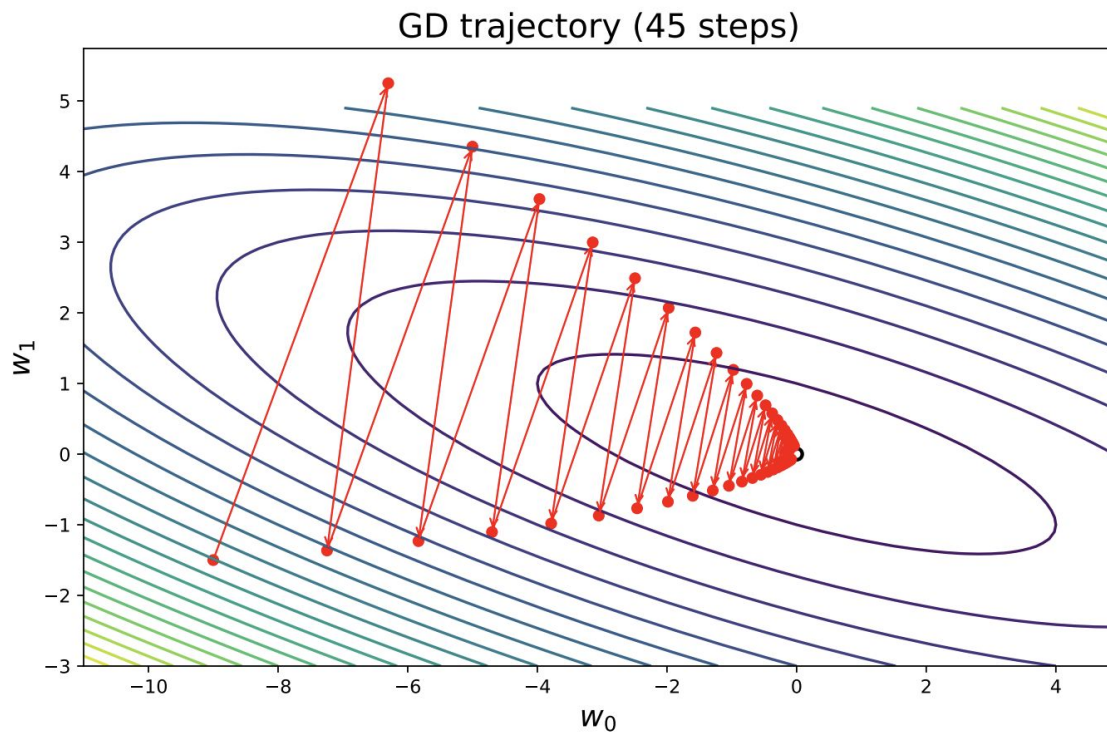
# Градиентный спуск: длина шага



# Градиентный спуск: длина шага



# Градиентный спуск: длина шага



## Градиентный спуск: переменная длина шага

$$x^t = x^{t-1} - \eta_t \nabla f(x^{t-1})$$

Длину шага можно менять в зависимости от итерации



# Градиентный спуск: переменная длина шага

$$x^t = x^{t-1} - \eta_t \nabla f(x^{t-1})$$

Длину шага можно менять в зависимости от итерации

- Например:  $\eta_t = \frac{1}{t}$
- Шаг наискорейшего спуска:

$$\eta_t = \arg \min_{\eta} f(x^t) = \arg \min_{\eta} f(x^{t-1} - \eta \nabla f(x^{t-1}))$$

# Градиентный спуск в МО

Хотим минимизировать ошибки модели на обучающей выборке

$$Q(w) = \frac{1}{l} \sum_{i=1}^l L(y_i, a(x_i))$$

$w^0$

1. Выбираем начальное приближение
2. На каждой итерации делаем шаг в сторону антиградиента

$$w = w - \eta \nabla Q(w)$$
$$\nabla Q(w) = \frac{1}{l} \sum_{i=1}^l \nabla L(y_i, a(x_i))$$

3. Останавливаемся, если  $\|Q(w^t) - Q(w^{t-1})\| < \varepsilon$

# Градиентный спуск: линейная регрессия

# Градиентный спуск: сложности

- Для вычисления градиента, как правило, надо просуммировать что-то по всем объектам

$$\nabla Q(w) = \frac{1}{l} \sum_{i=1}^l \nabla L(y_i, a(x_i))$$

- И это для одного маленького шага!

# Градиентный спуск: сложности

- Для вычисления градиента, как правило, надо просуммировать что-то по всем объектам

$$\nabla Q(w) = \frac{1}{l} \sum_{i=1}^l \nabla L(y_i, a(x_i))$$

- И это для одного маленького шага!
- Может оценить одним слагаемым?

$$\nabla Q(w) \approx \nabla L(y_i, a(x_i))$$

# Стохастический градиентный спуск

Хотим минимизировать ошибки модели на обучающей выборке

$$Q(w) = \frac{1}{l} \sum_{i=1}^l L(y_i, a(x_i))$$

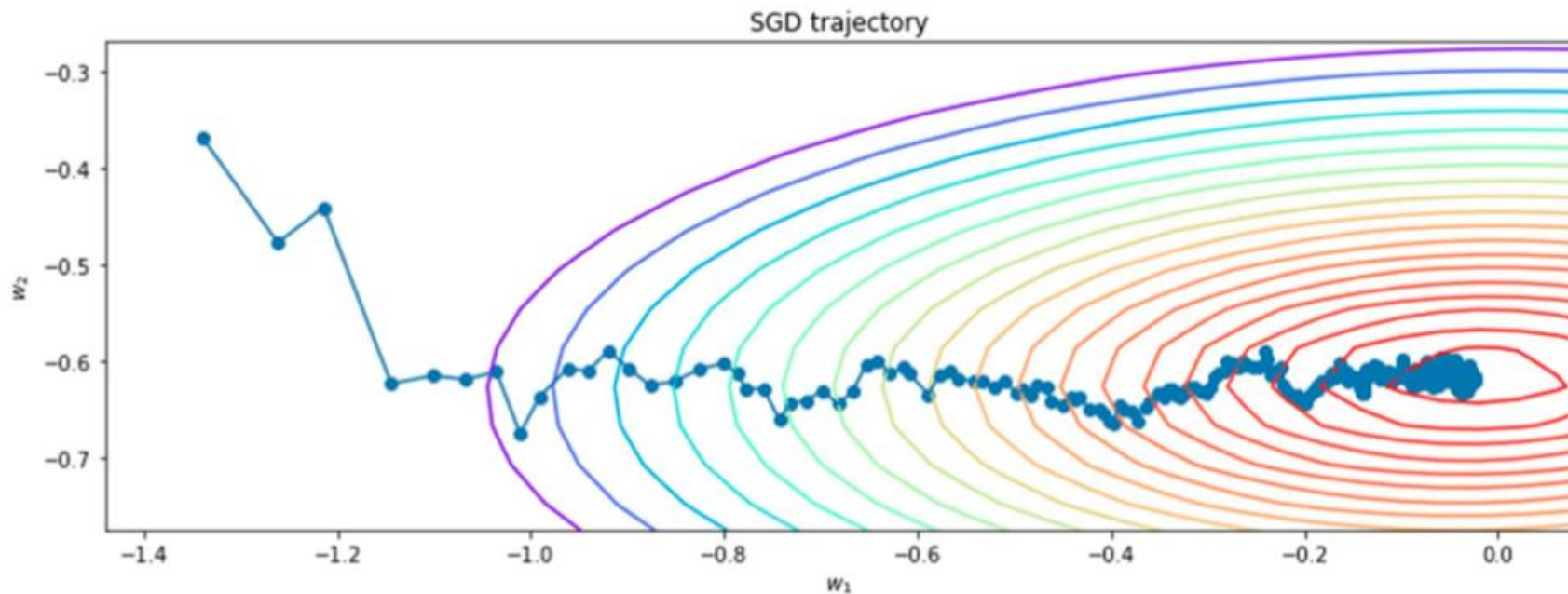
$w^0$

1. Выбираем начальное приближение
2. На каждой итерации выбираем случайный объект  $i$  и делаем шаг

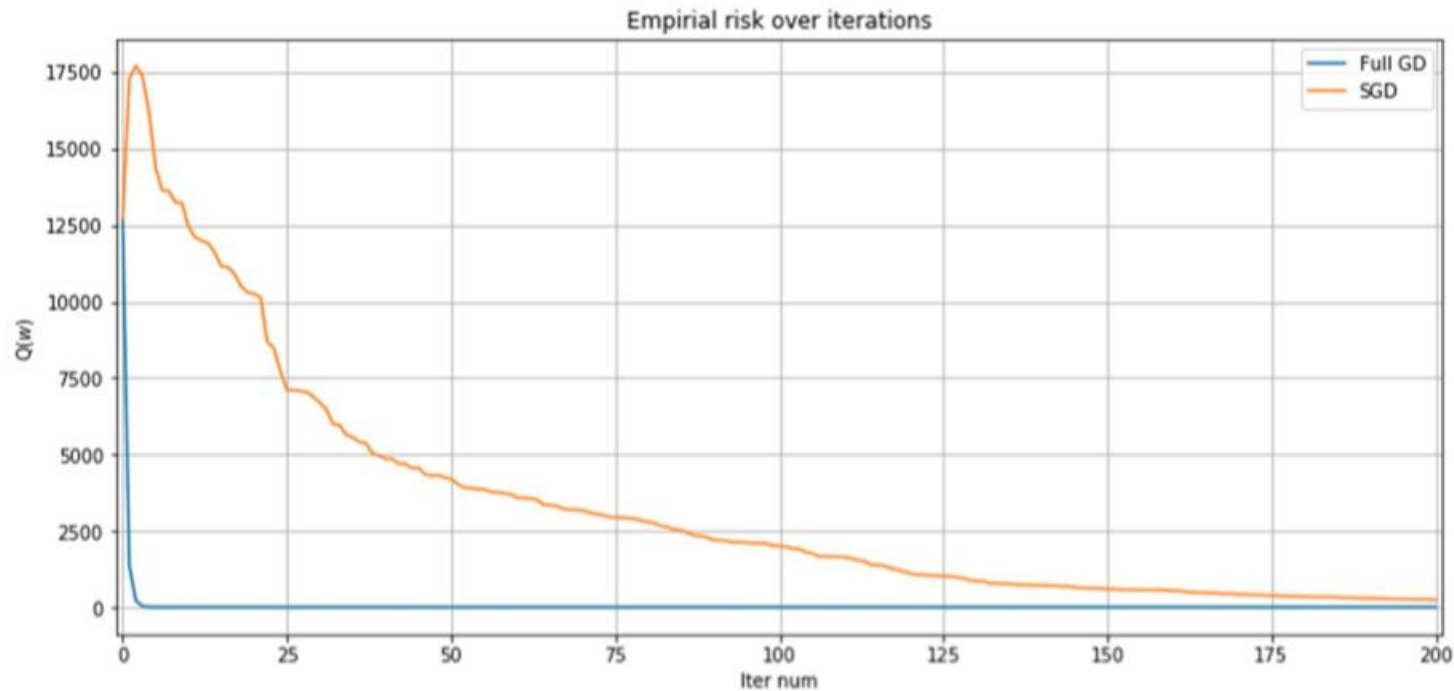
$$w^t = w^{t-1} - \eta \nabla L(y_i, a(x_i))$$

3. Останавливаемся, если  $\|Q(w^t) - Q(w^{t-1})\| < \varepsilon$

# Стохастический градиентный спуск



# Стохастический градиентный спуск





# Mini-batch

Хотим минимизировать ошибки модели на обучающей выборке

$$Q(w) = \frac{1}{l} \sum_{i=1}^l L(y_i, a(x_i))$$

$w^0$

1. Выбираем начальное приближение
2. На каждой итерации выбираем  $m$  случайных объектов и делаем шаг  $w^t = w^{t-1} - \eta \frac{1}{m} \sum_{j=1}^m \nabla L(y_j, a(x_j))$
3. Останавливаемся, если  $\|Q(w^t) - Q(w^{t-1})\| < \varepsilon$

# Градиентный спуск

1. Mini-batch GD стабильнее стохастического
2. Важно масштабировать признаки
3. Длина шага - гиперпараметр, который сильно влияет на сходимость