

# MML minor #1

Методы оптимизации

# Задача машинного обучения

$D = \{x, y\}_{i=1}^N$  — обучающая выборка.

$y^* = A(x; \mu)$  — метод предсказания  
 $\mu$  — параметры

$L(D, \mu) = E(D, \mu) + R(\mu)$  — функция потерь  
 $E(D, \mu)$  — функция ошибки  
 $R(\mu)$  — функция регуляризации

$L(D, \mu) \rightarrow \min_{\mu}$  — процедура обучения

Ключевая задача в машинном обучении: **оптимизация**.

# Напоминание: ряд Тейлора

$f(x)$  — одномерная бесконечно дифференцируемая функция

$$f(x) = f(x_0) + \frac{f'(x_0)}{1!}(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \dots =$$

$$= \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n.$$

(не всегда сходится)

# Напоминание: градиент

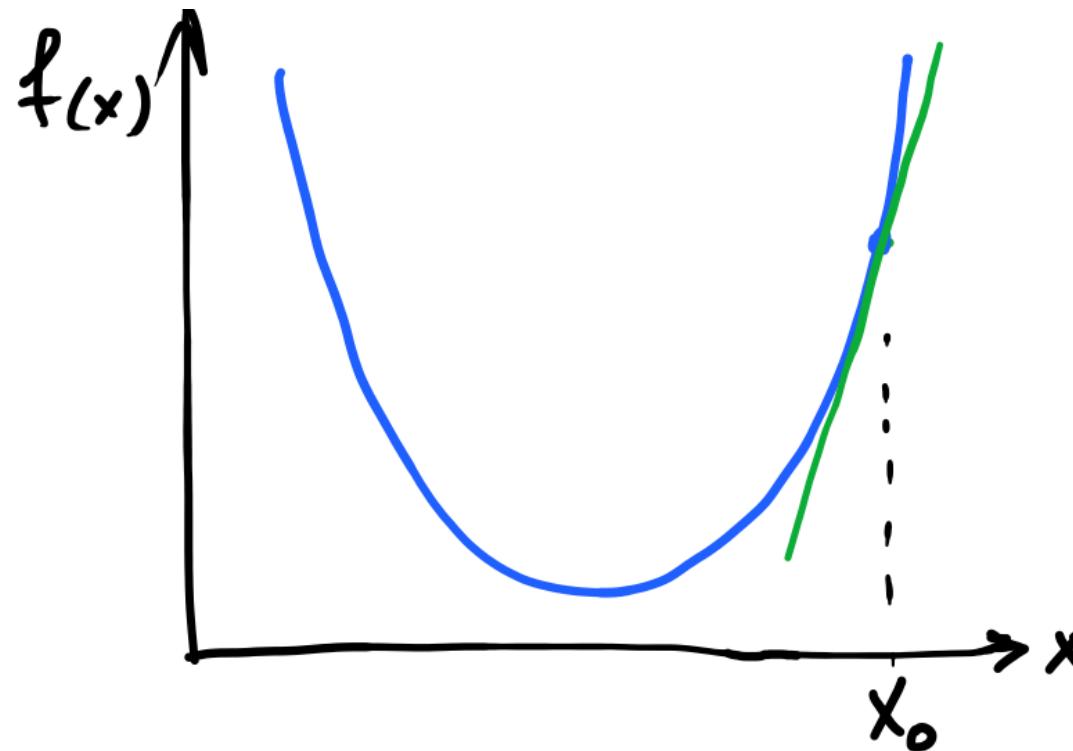
- Пусть  $f = f(x, y)$
- Частная производная:  $\frac{\partial f}{\partial x}$
- Градиент:  $\nabla f = \left( \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)$

# Методы 1 порядка

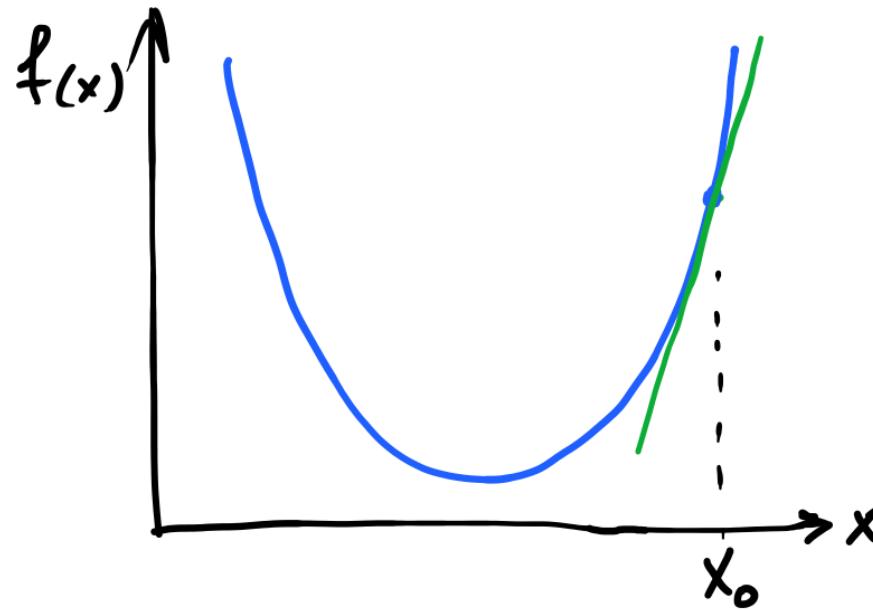
$$f(x) \rightarrow \min_x$$

Разложим в ряд Тейлора до первой производной

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0)$$



# Градиент как направление шага



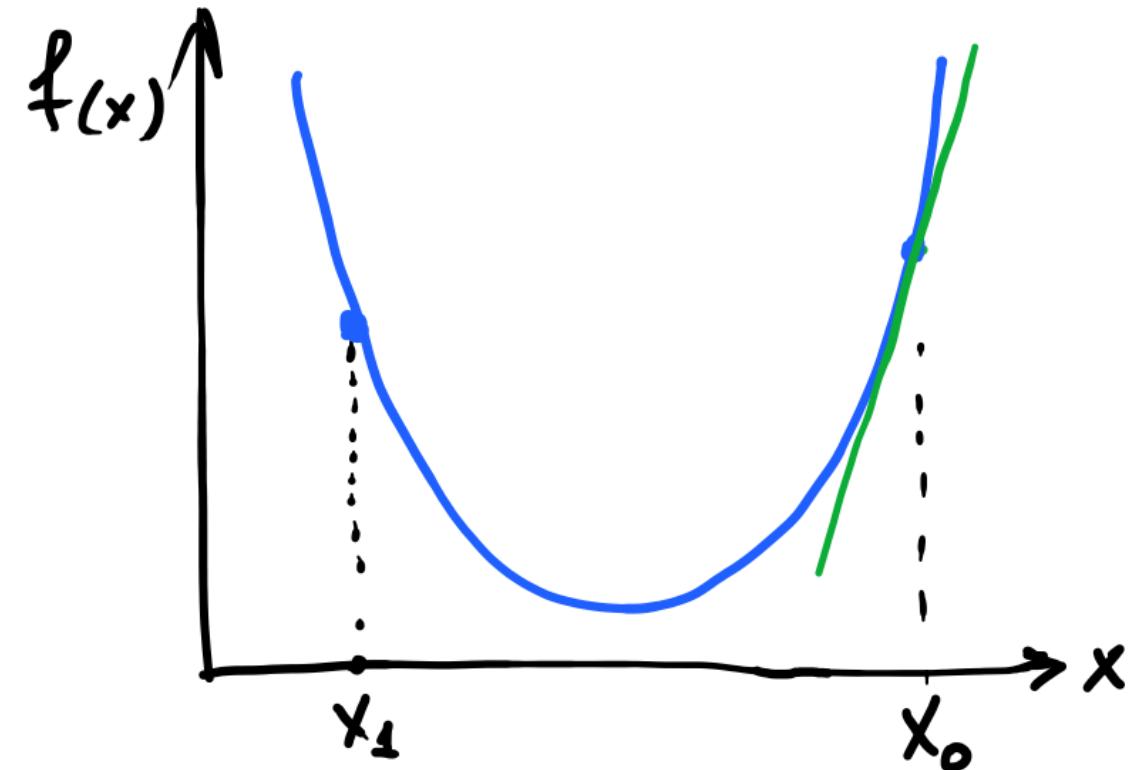
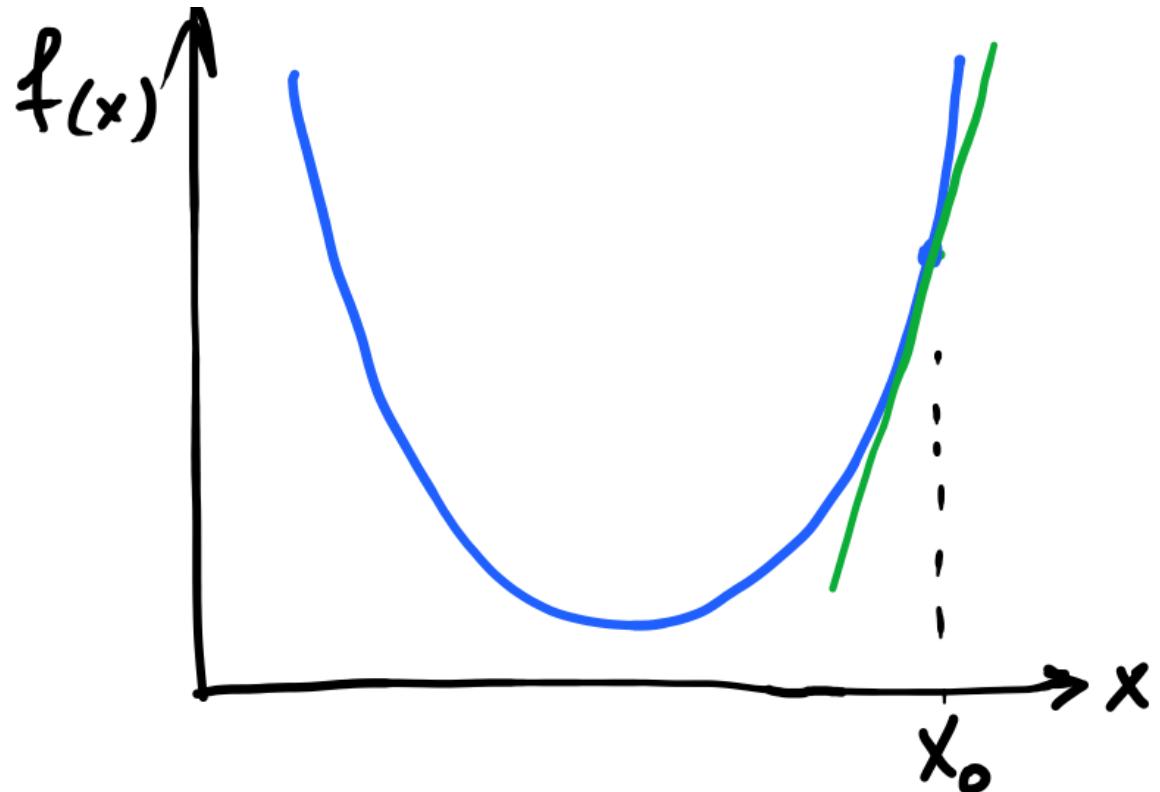
$$f(x_0) + f'(x_0)(x - x_0) \rightarrow \min_x$$

$f'(x_0) > 0$  — минимум в направлении  $x \rightarrow -\infty$

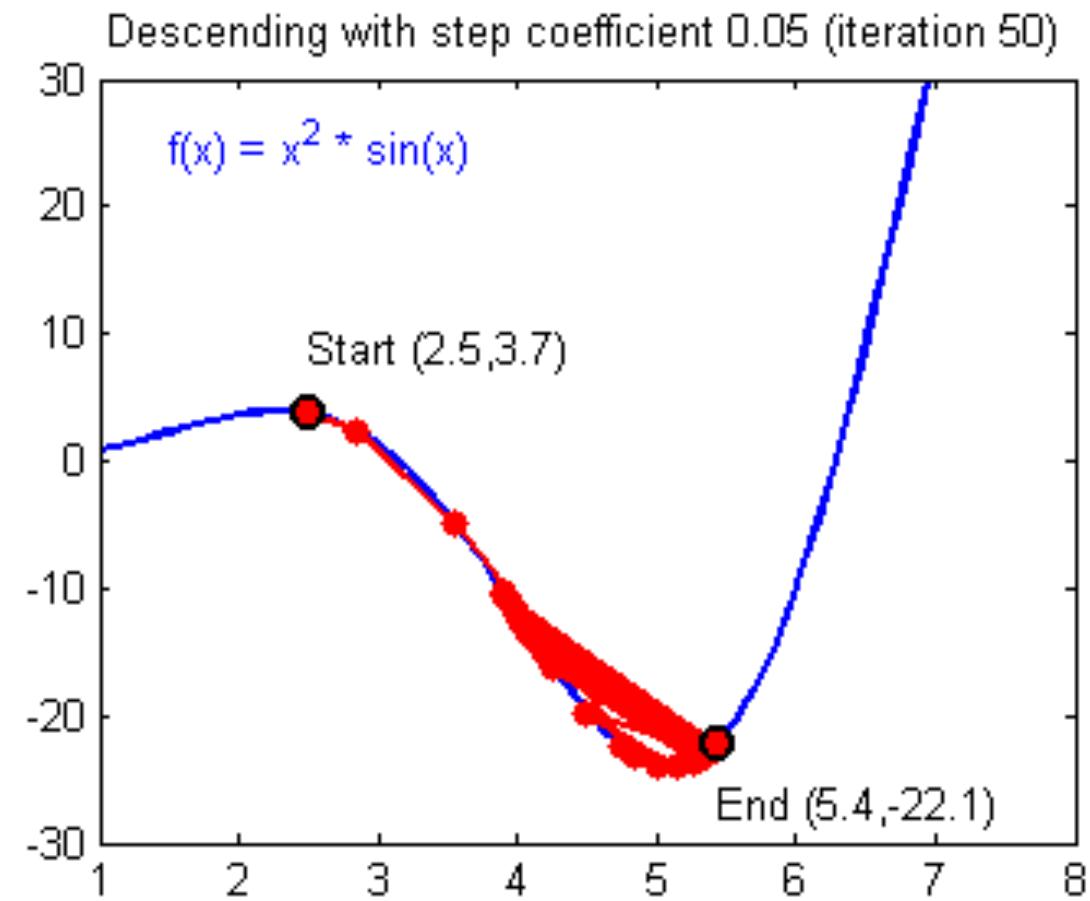
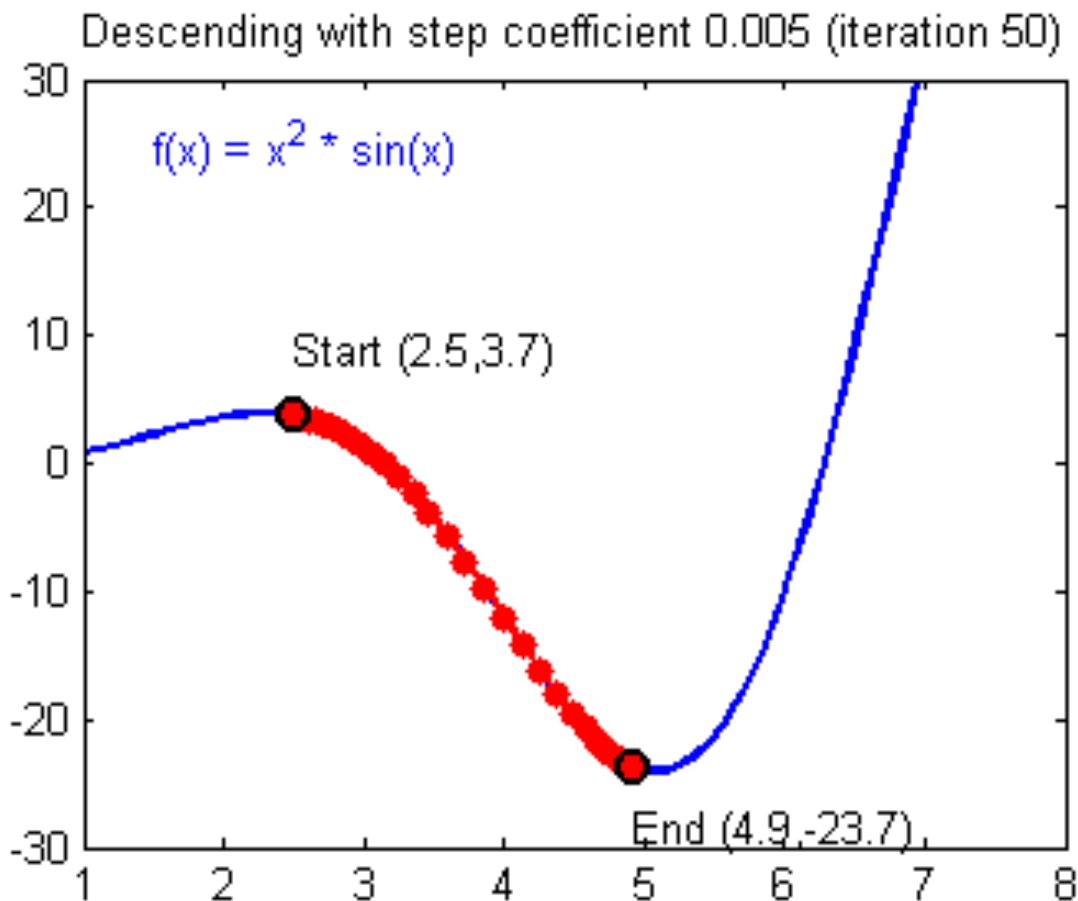
$f'(x_0) < 0$  — минимум в направлении  $x \rightarrow +\infty$

Оптимальный шаг: в направлении  $-f'(x_0)$

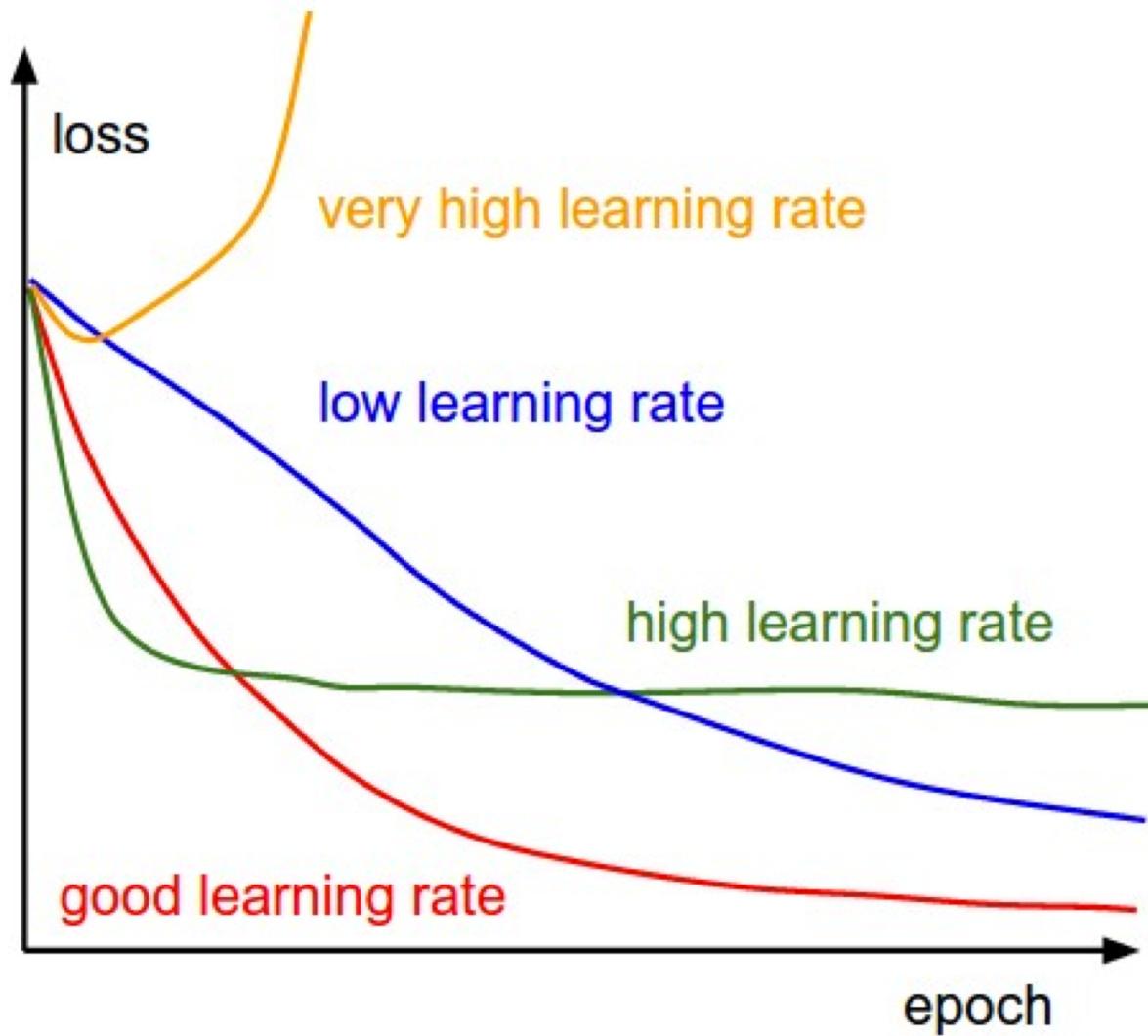
# Величина шага?



# Величина шага?



# Как понять что величина шага плохая



# Многомерный случай

Задача:

$$f(x) \rightarrow \min_x$$

Разложим в ряд Тейлора до первой производной

$$f(x) \approx f(x_0) + (x - x_0)^T \nabla f(x_0)$$

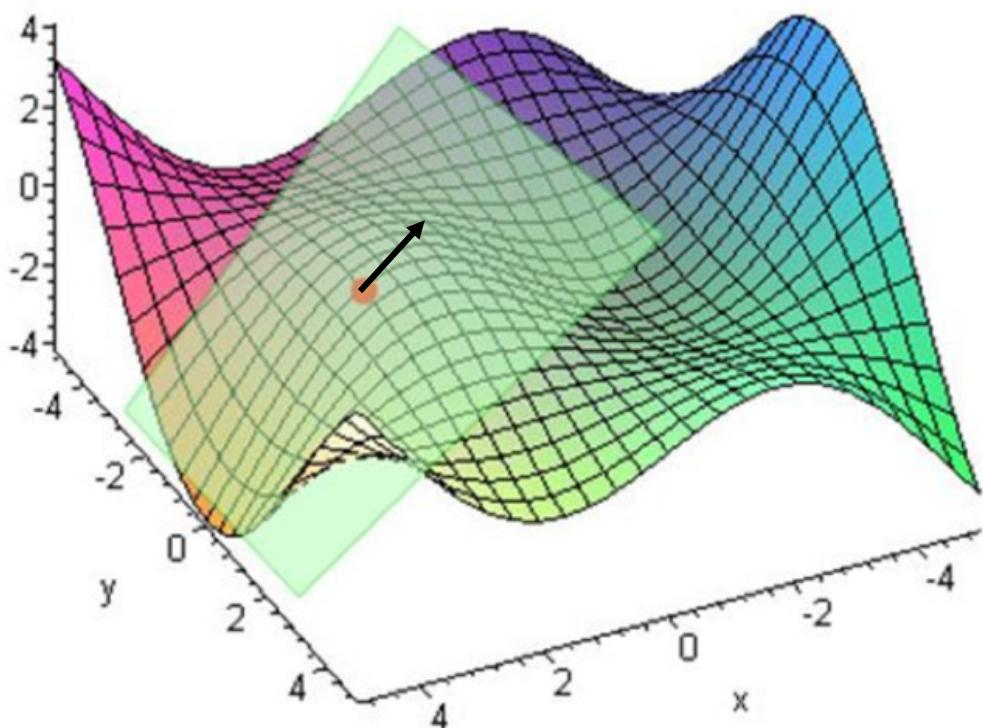
Когда достигается максимум  $\langle x - x_0, \nabla f(x_0) \rangle$ ?

Ответ: когда  $(x - x_0)$  и  $\nabla f(x_0)$  параллельны

Наибольшее возрастание:  $x - x_0 = \eta \nabla f(x_0)$

Наибольшее убывание:  $x - x_0 = -\eta \nabla f(x_0)$

# Многомерный случай



$$f(a) \approx f(x_0) + (a - x_0)^T \nabla f(x_0)$$

$$f(b) \approx f(x_0) + (b - x_0)^T \nabla f(x_0)$$

$$\Delta z = f(a) - f(b) = \frac{\partial f}{\partial x} \Delta x + \frac{\partial f}{\partial y} \Delta y$$

Мы знаем как меняется функция  
при небольшом изменении аргументов

Для максимизации выгодно пойти вдоль  $\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)$

# Градиентный спуск (GD)

$$f(x) \rightarrow \min_x$$

$\eta$  — величина шага (гиперпараметр)

① Инициализация

$$k = 0, \quad x_k = \text{начальное приближение}$$

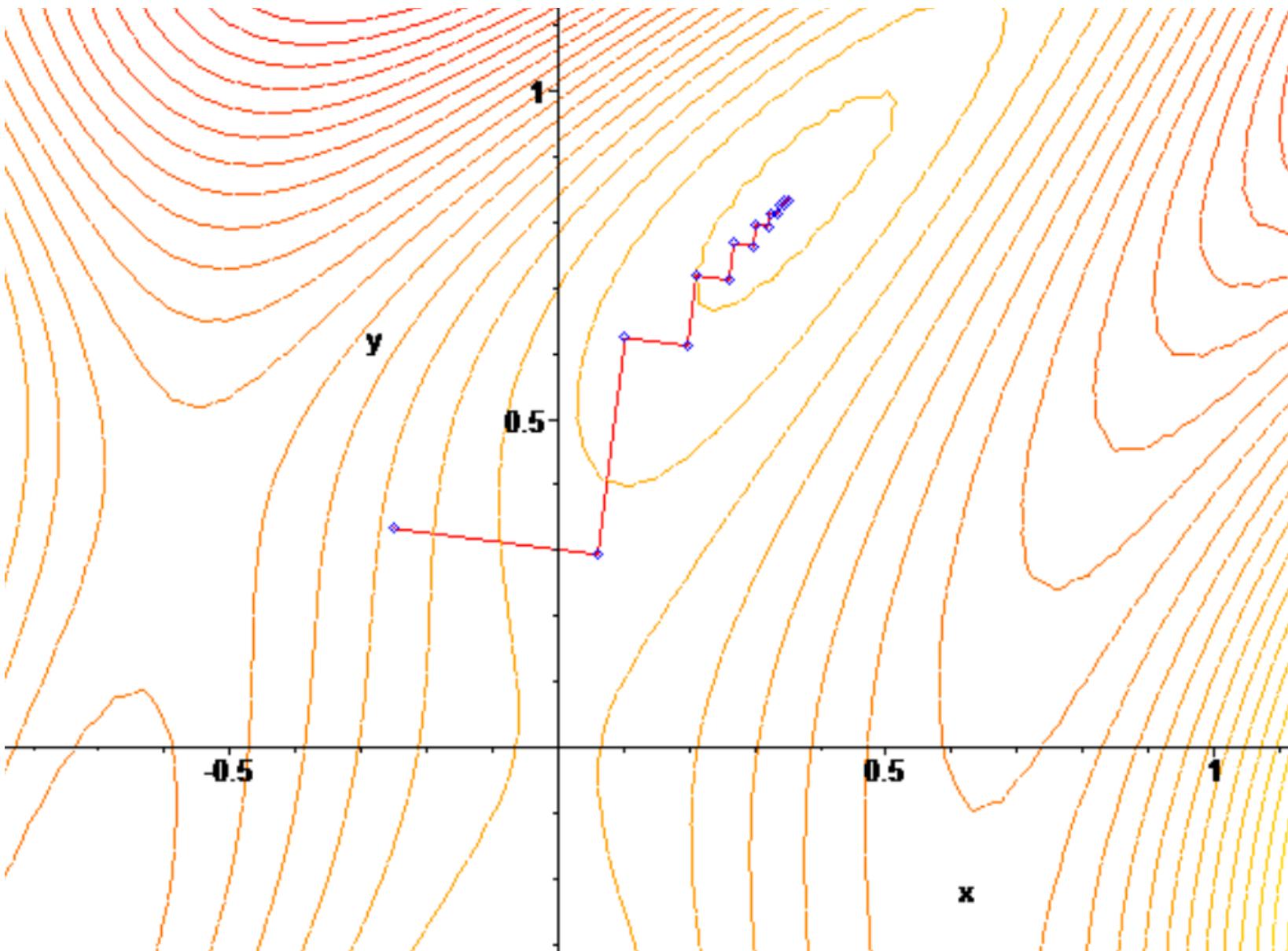
② Шаг в сторону сильнейшего убывания

$$x_{k+1} = x_k - \eta \nabla f(x_k)$$

③ Повторение до сходимости

$$k := k + 1, \quad \text{перейти к 2}$$

# Градиентный спуск: шаги



# Градиентный спуск

## Плюсы

- Достаточно универсальный метод
- Легко реализуем

## Минусы

- Попадает в локальные минимумы
- Шаги могут быть медленными
- Неэффективен для больших выборок
- Неприменим для недифференцируемых функций

# Застревает в локальных минимумах



Bob chillin at a local optima

# Стохастический градиент

Пусть  $D$  содержит очень много элементов.

- Долго считать градиент
- Можно сделать мало итераций за разумное время
- Плохое решение

Идея: приближенно оценивать градиент.

# Стохастический градиент

Разложим функцию потерь:

$$L(D, \mu) = \frac{1}{N} \sum_{i=1}^N l(x_i, \mu).$$

$$\nabla_\mu L(D, \mu) = \frac{1}{N} \sum_{i=1}^N \nabla_\mu l(x_i, \mu).$$

Пусть  $I = (i_1, i_2, \dots, i_m)$  — небольшая подвыборка  $D$

Приблизим градиент:

$$\nabla_\mu L(D, \mu) \approx \frac{1}{m} \sum_I \nabla_\mu l(x_i, \mu).$$

# Стохастический градиентный спуск (SGD)

$$\sum_{i=1}^N f_i(x) \rightarrow \min_x$$

$\eta$  — величина шага,  $m$  — размер подвыборки

① Инициализация

$$k = 0, \quad x_k = \text{начальное приближение}$$

② Шаг *примерно* в сторону сильнейшего убывания

$I :=$  случайная подвыборка размера  $m$

$$x_{k+1} = x_k - \eta \sum_I \nabla f_i(x_k)$$

③ Повторение до сходимости

$$k := k + 1, \quad \text{перейти к 2}$$

# SGD mini-batches

На каждой итерации:

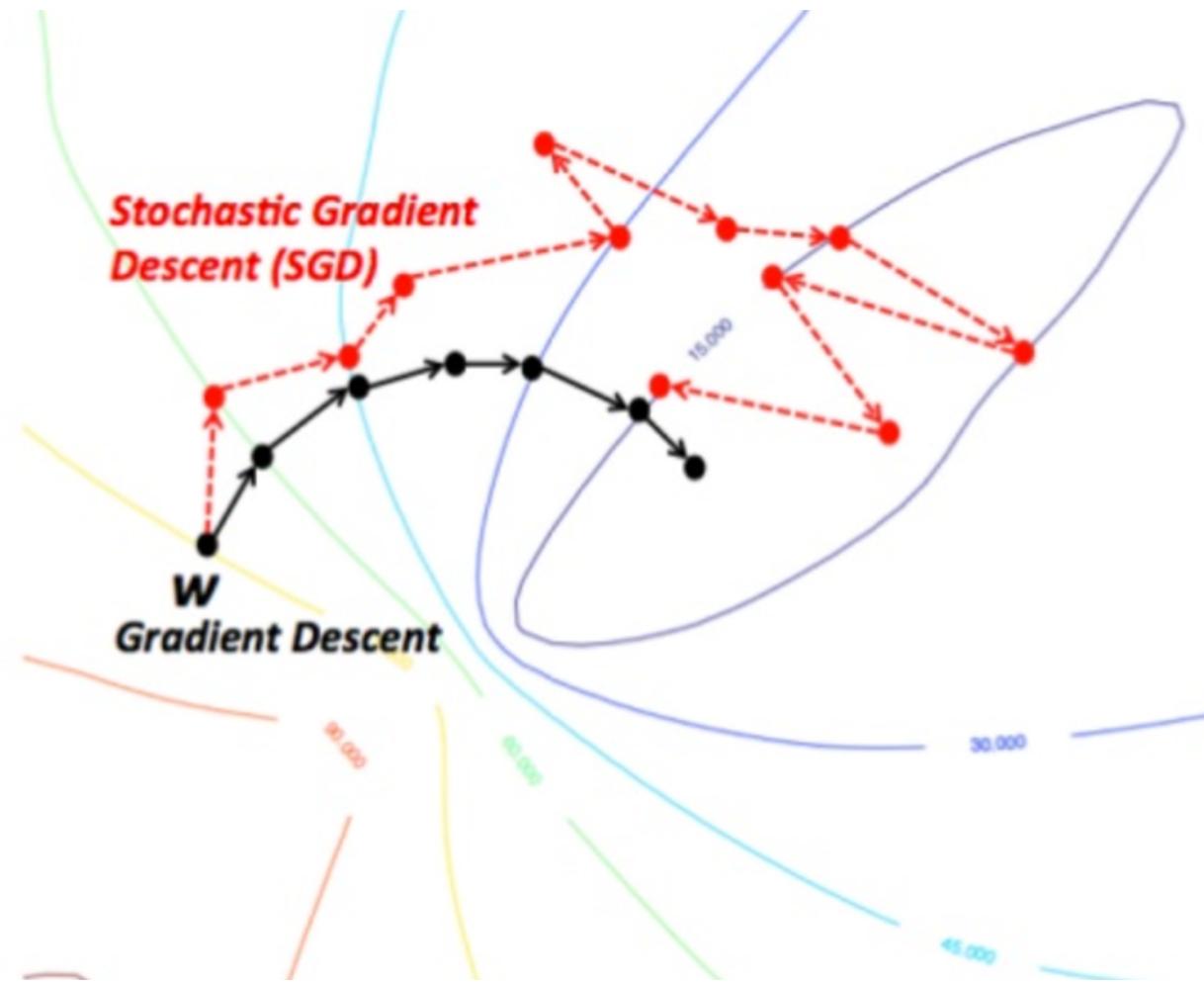
- Перемешать выборку
- Разбить выборку на равные части размера  $m$  (мини-батчи)

$$I_1 + I_2 + \dots + I_{N/m} = \{1, \dots, N\}$$

- Для  $j = 1, \dots, N/m$

$$x := x - \eta \sum_{I_j} \nabla f_i(x)$$

# SGD: шаги



Шум в оценке градиента помогает выпрыгивать из локальных оптимумов

# (S)GD + инерция (momentum)

$$f(x) \rightarrow \min_x$$

$\eta$  — величина шага,  $\alpha$  — влияние инерции

① Инициализация

$$k = 0, \quad x_k = \text{начальное приближение}, \quad m_k = 0$$

② Шаг в сторону сильнейшего убывания

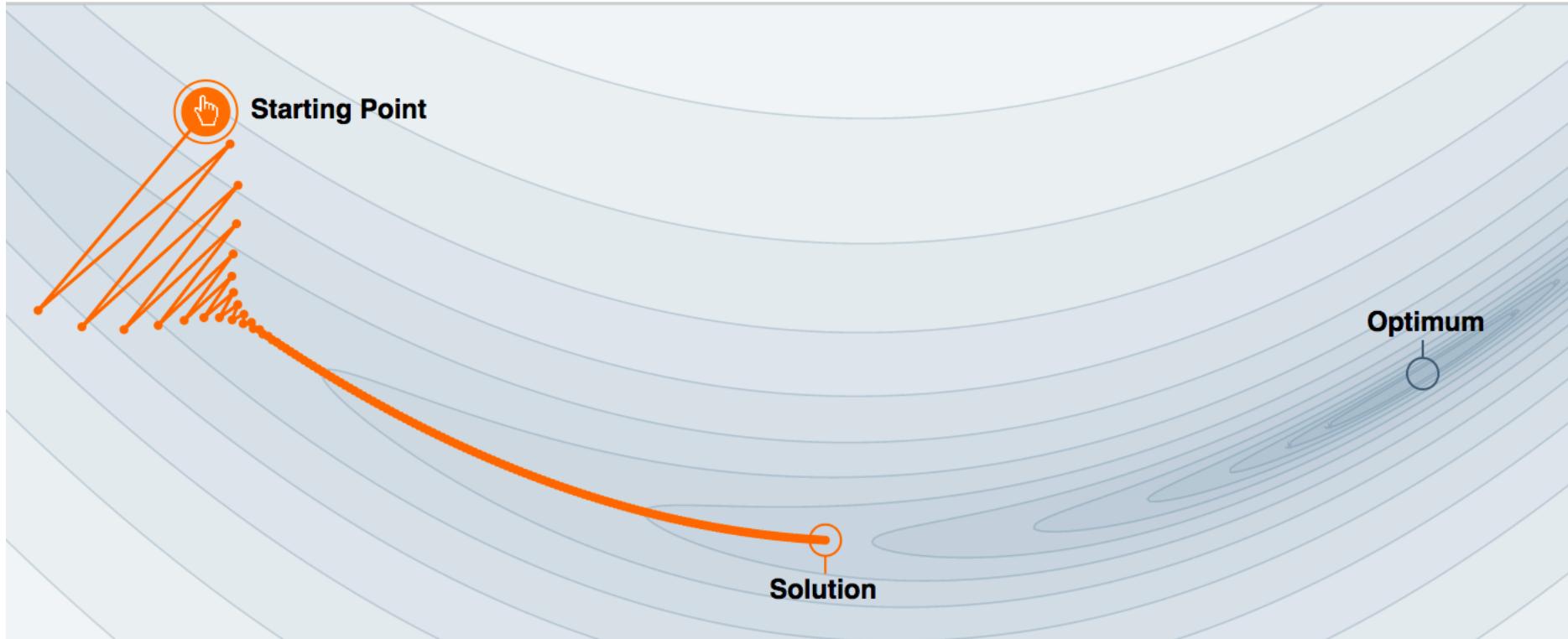
$$m_{k+1} := -\eta \nabla f(x_k) + \alpha m_k$$

$$x_{k+1} = x_k + m_{k+1}$$

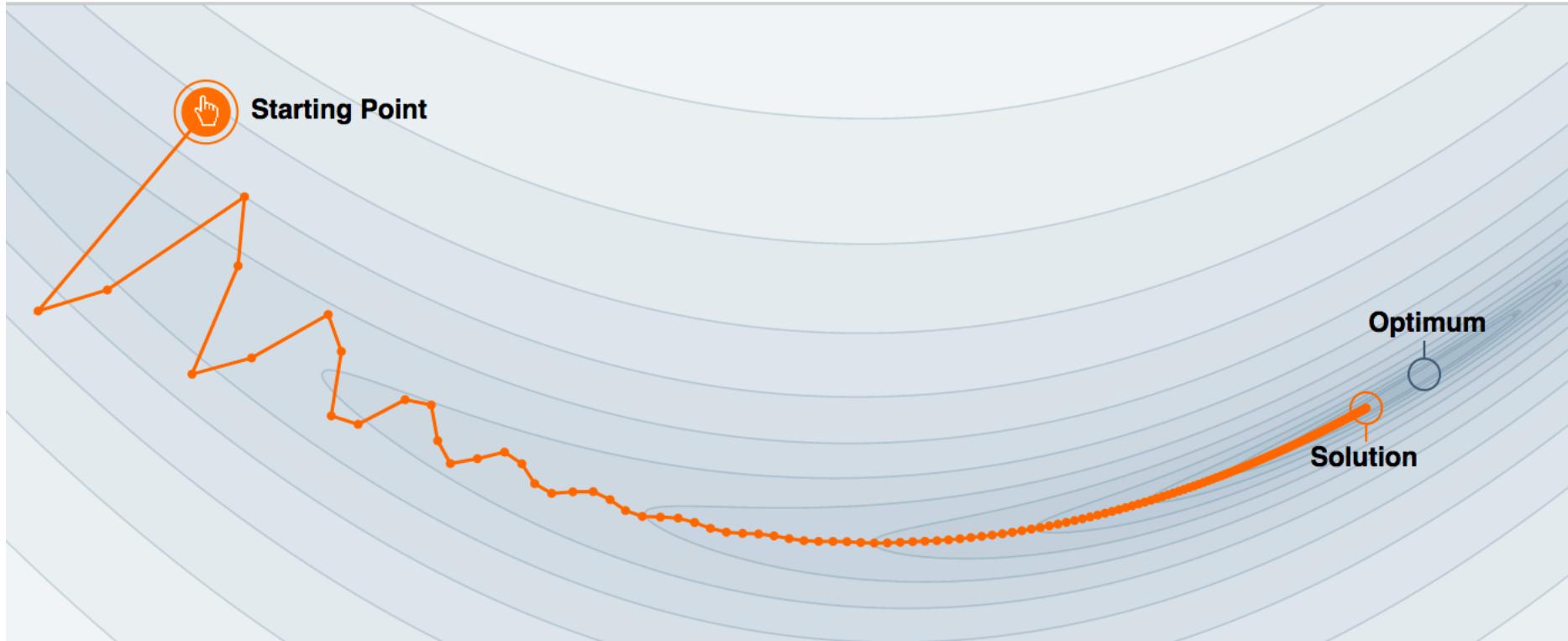
③ Повторение до сходимости

$$k := k + 1, \quad \text{перейти к 2}$$

# Сходимость без инерции



# Сходимость с инерцией



«Тяжелый мячик катится по поверхности»

# GD + линейный поиск

$$f(x) \rightarrow \min_x$$

① Инициализация

$$k = 0, \quad x_k = \text{начальное приближение}$$

② Минимизация в направлении сильнейшего убывания

$$\alpha^* = \arg \min_{\alpha} f(x_k - \alpha \nabla f(x_k))$$

$$x_{k+1} = x_k - \alpha^* \nabla f(x_k)$$

③ Повторение до сходимости

$$k := k + 1, \quad \text{перейти к 2}$$

# (S)GD + переменный шаг

$$f(x) \rightarrow \min_x$$

$\eta$  — базовая величина шага (гиперпараметр)

- ① Инициализация

$k = 0, \quad x_k = \text{начальное приближение}$

- ② Шаг в сторону сильнейшего убывания

$$x_{k+1} = x_k - \frac{\eta}{k} \nabla f(x_k)$$

- ③ Повторение до сходимости

$k := k + 1, \quad \text{перейти к 2}$

# Методы 2-го порядка

$$f(x) \rightarrow \min_x$$

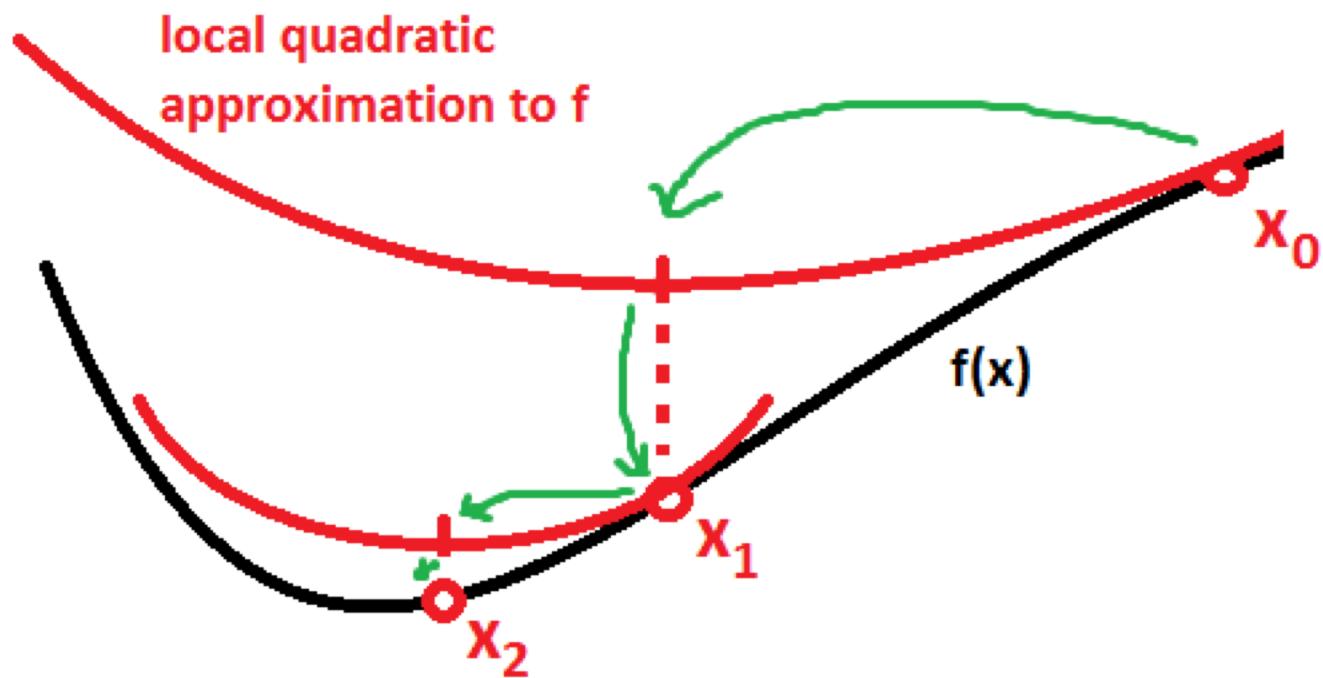
Разложим в ряд Тейлора до второй производной

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2$$

Минимизируем

$$f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2 \rightarrow \min_x$$

# Методы 2-го порядка



# Методы 2-го порядка

Минимизируем

$$f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2 \rightarrow \min_x$$

Минимум при

$$x = x_0 - \frac{f'(x_0)}{f''(x_0)}$$

# Многомерный случай

$$f(x) \rightarrow \min_x$$

Разложим в ряд Тейлора до второй производной

$$f(x) \approx f(x_0) + (x - x_0)^T \nabla f(x_0) + \frac{1}{2} (x - x_0)^T \nabla^2 f(x_0) (x - x_0)$$

Минимизируем

$$f(x_0) + (x - x_0)^T \nabla f(x_0) + \frac{1}{2} (x - x_0)^T \nabla^2 f(x_0) (x - x_0) \rightarrow \min_x$$

Минимум при

$$x = x_0 - (\nabla^2 f(x_0))^{-1} \nabla f(x_0)$$

# Метод Ньютона

$$f(x) \rightarrow \min_x$$

$\eta$  — величина шага (гиперпараметр)

- ① Инициализация

$k = 0, \quad x_k = \text{начальное приближение}$

- ② Шаг в сторону примерного минимума

$$x_{k+1} = x_k - \eta \boxed{(\nabla^2 f(x_k))^{-1}} \nabla f(x_k)$$

- ③ Повторение до сходимости

$k := k + 1, \quad \text{перейти к 2}$

# Метод Ньютона

- Плюсы:
  - Сходится за меньшее число шагов по сравнению с GD
- Минусы:
  - Долго вычислять Якобиан  $\nabla^2 f(x_k)$

На практике используют квази-Ньютоновские методы, такие как L-BFGS

# Методы 0 порядка

(Производную нельзя вычислить)

$$f(x) = f(x_1, x_2, \dots, x_n) \rightarrow \min_x$$

Минимизируем вдоль одной координаты  $i$

$$f(x_1, \dots, x_{i-1}, z, x_{i+1}, \dots, x_n) \rightarrow \min_z$$

Минимизация линейным поиском  
(перебор всех значений в окрестности).

# Покоординатный спуск

$$f(x) = f(x_1, x_2, \dots, x_n) \rightarrow \min_x$$

① Инициализация

$$k = 0, \quad x^k = \text{начальное приближение}$$

② Минимизируем вдоль координаты  $i$  для  $i = 1, \dots, n$

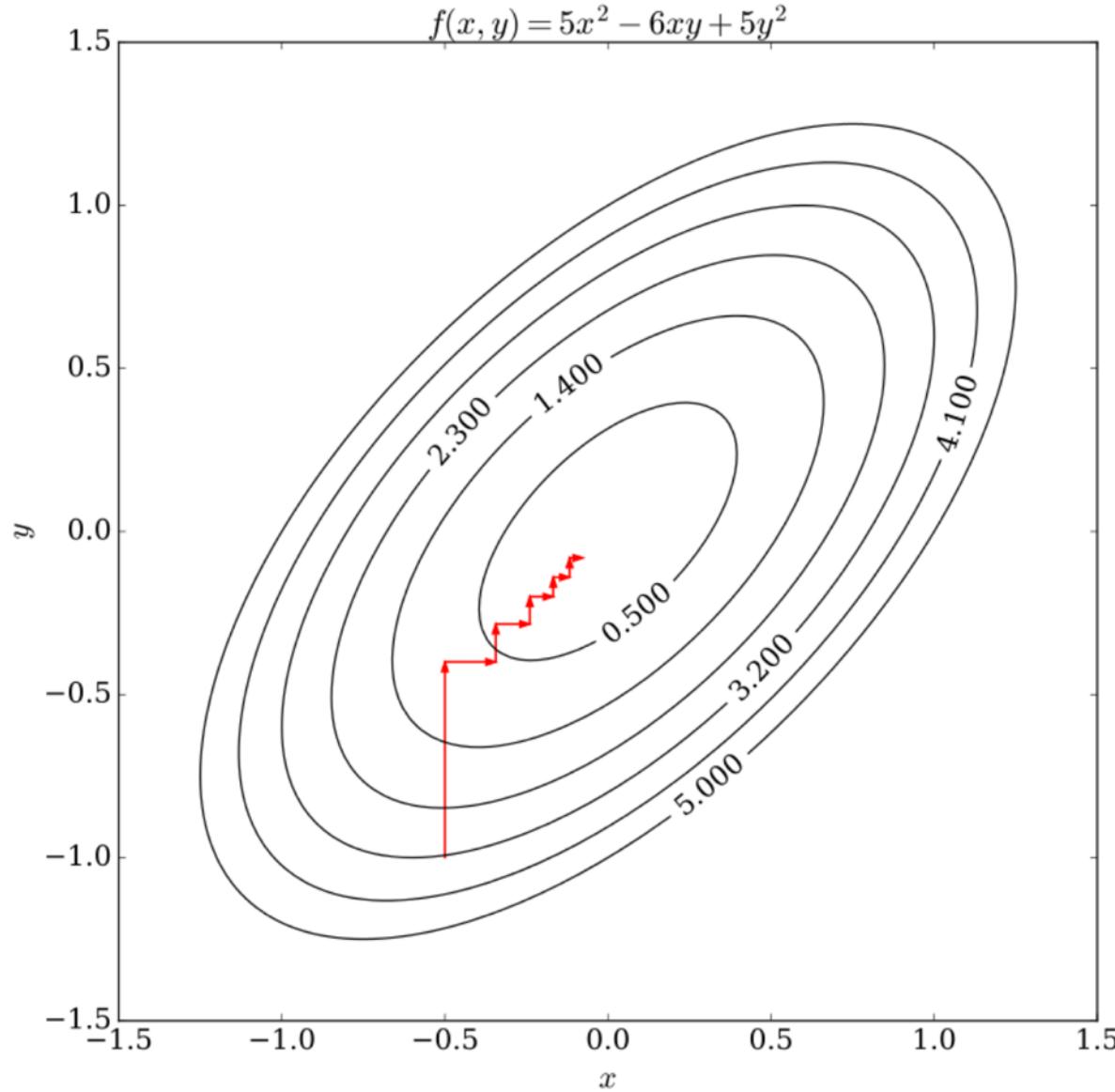
$$z^* = \arg \min_z f(x_1, \dots, x_{i-1}, z, x_{i+1}, \dots, x_n)$$

$$x^{k+1} = (x_1^k, \dots, x_{i-1}^k, z^*, x_{i+1}^k, \dots, x_n^k)$$

$$k := k + 1$$

③ Повторение до сходимости

# Покоординатный спуск: шаги



# Покоординатный спуск

- Плюсы:
  - Не нужно вычислять производную
- Минусы:
  - Долго работает

# Ссылки

- <https://hackernoon.com/life-is-gradient-descent-880c60ac1be8>
- <https://distill.pub/2017/momentum/>
- <http://slideplayer.com/slide/7341917/> Лекции по оптимизации
- <http://ruder.io/optimizing-gradient-descent/>