Методы оптимизации в машинном обучении.

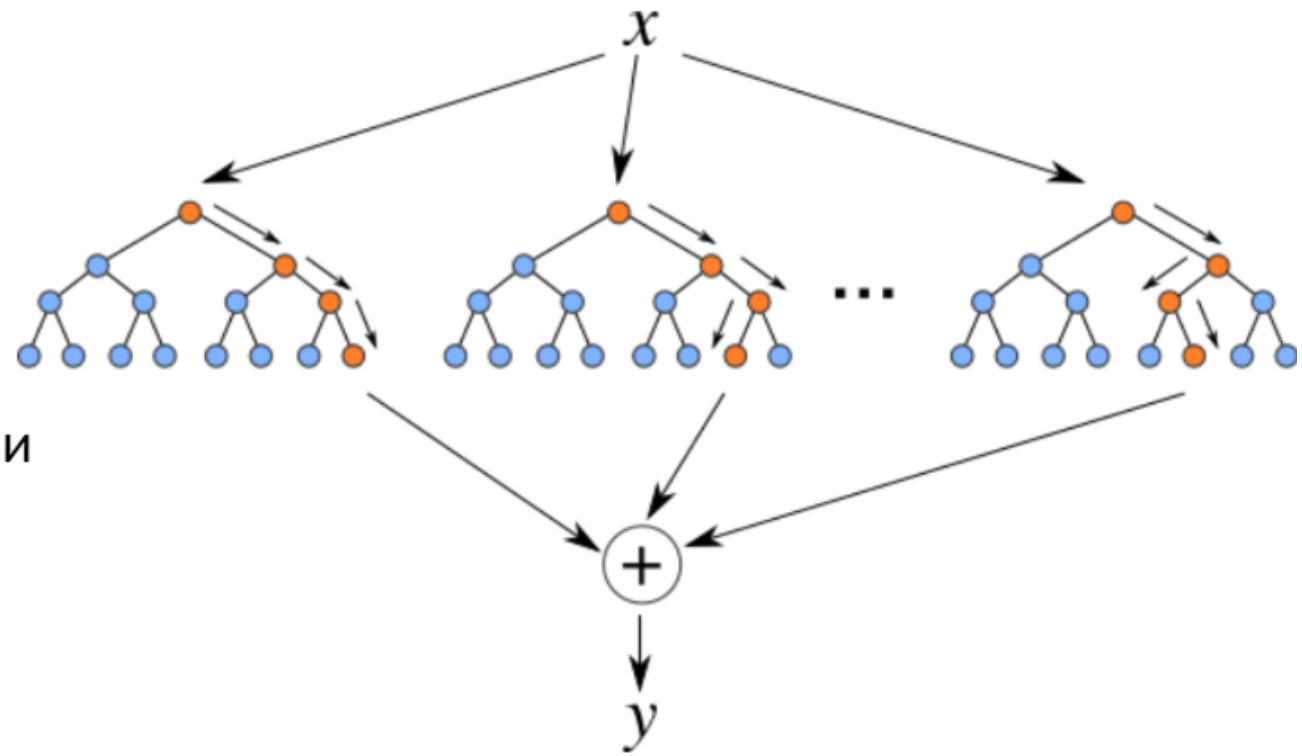
Градиент

Random Forest

1. Генерируем M выборок на основе имеющейся (по схеме выбора с возвращением)

2. Строим на них деревья с рандомизированными разбиениями в узлах: выбираем k случайных признаков и ищем наиболее информативное разбиение по ним

3. При прогнозе усредняем ответ всех деревьев



Градиент математика

$$f'(x_0) = \lim_{x \to x_0} \frac{f(x) - f(x_0)}{x - x_0} = \lim_{\Delta x \to 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} = \lim_{\Delta x \to 0} \frac{\Delta f(x)}{\Delta x}.$$

Поиск минимума функции

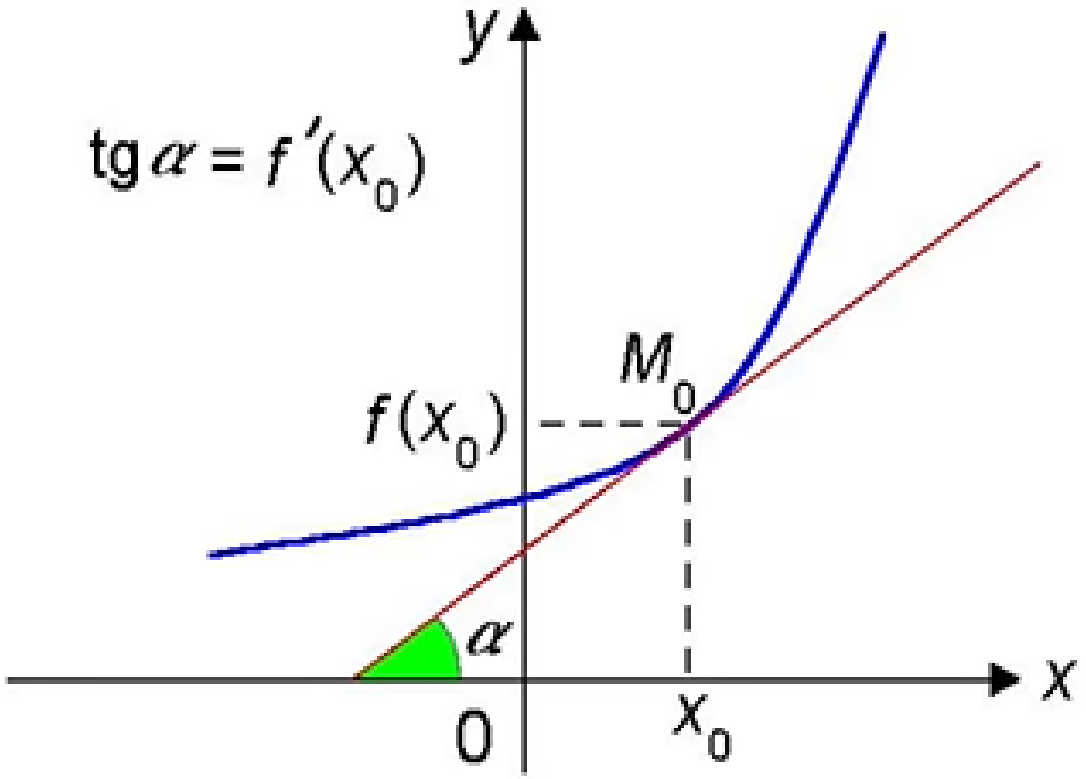
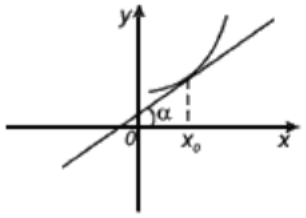
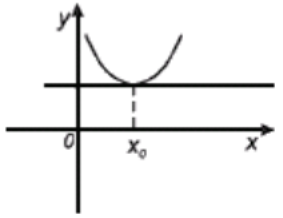


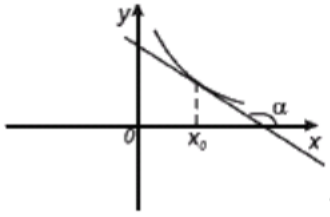
Рис. 1



$f'(x_0) = tg\alpha > 0$

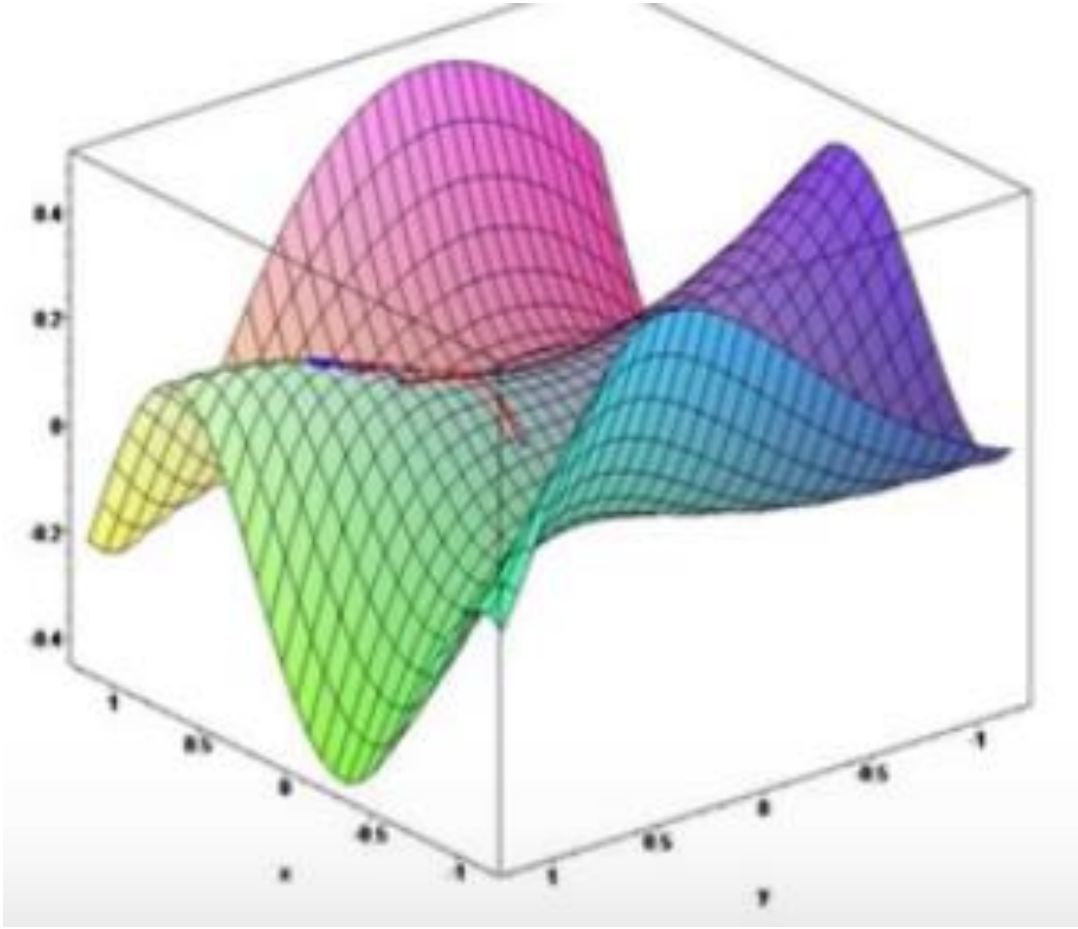


$f'(x_0) = tg\alpha = 0$



$f'(x_0) = tg\alpha < 0$

Многомерный случай



Понятие частной производной

Частная производная — это одно из обобщений понятия производной на случай функции нескольких переменных.

Частная производная функции $f(x, y)$ по x определяется как производная по x, взятая в смысле функции одной переменной, при условии постоянства оставшейся переменной y. Таким образом:

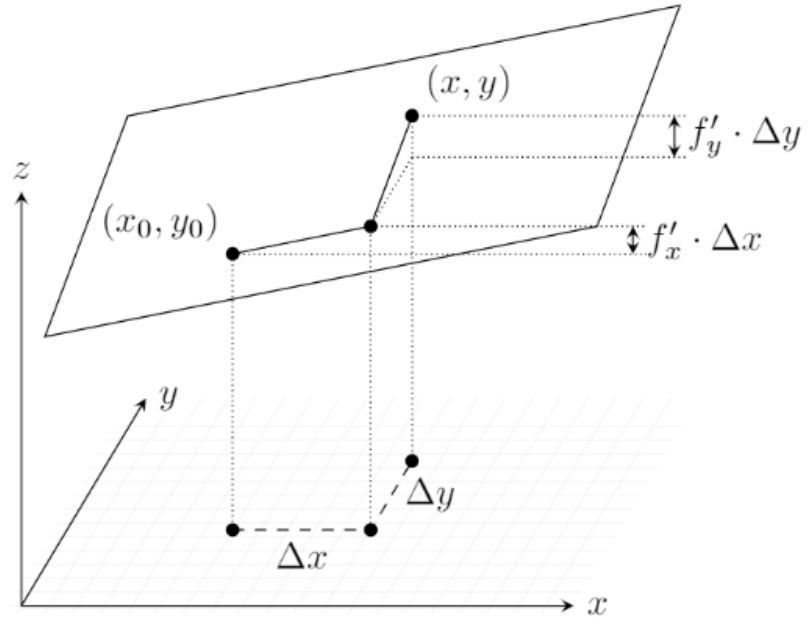
$$f'_x(x, y) = \lim_{h \to 0} \frac{f(x + h, y) - f(x, y)}{h}, \qquad f'_y(x, y) = \lim_{h \to 0} \frac{f(x, y + h) - f(x, y)}{h}.$$

Геометрический смысл частной производной

Пусть дана некоторая функция двух переменных $f : \mathbb{R}^2 \to \mathbb{R}$. Она, вообще говоря, определяет некоторую поверхность $z = f(x, y)$ в трехмерном пространстве.

Если в некоторой точке (x_0, y_0) функция дифференцируема как функция многих переменных, то в этой точке можно рассмотреть плоскость касательную к рассматриваемой поверхности.

Эта касательная плоскость пересекает координатные плоскости $x0z$ и $y0z$ по прямым, тангенсы угла между которыми и соответствующими координатными осями равны значениям частных производных в точке (x_0, y_0).



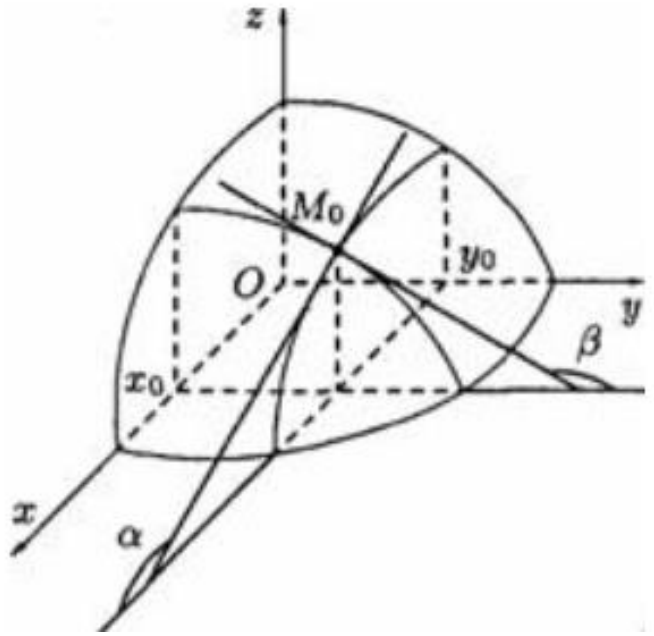


Рис. 1: Геометрический смысл частных производных.

Таким образом, график функции $f(x, y)$ в окрестности точки можно приблизить касательной плоскостью:

$$f(x_0 + \Delta x, y_0 + \Delta y) \approx f(x_0, y_0) + f'_x(x_0, y_0)\Delta x + f'_y(x_0, y_0)\Delta y.$$

Градиент

Если $f(x_1, \ldots, x_n)$ — функция n переменных $x_1, ..., x_n$, то n-мерный вектор из частных производных:

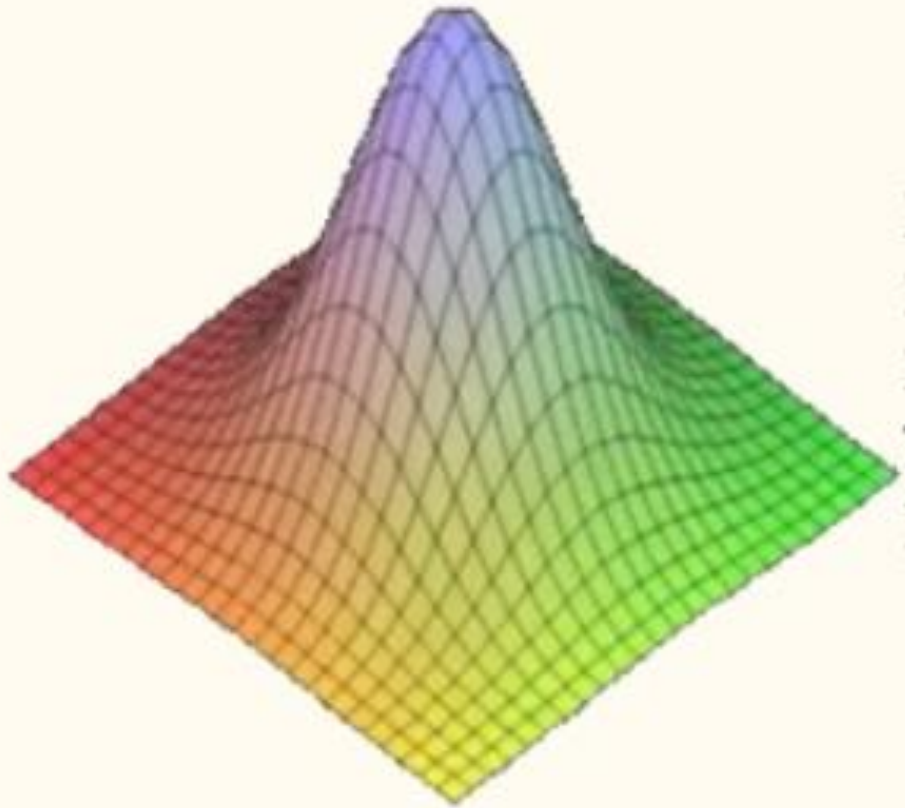
$$\operatorname{grad} f = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n} \right)$$

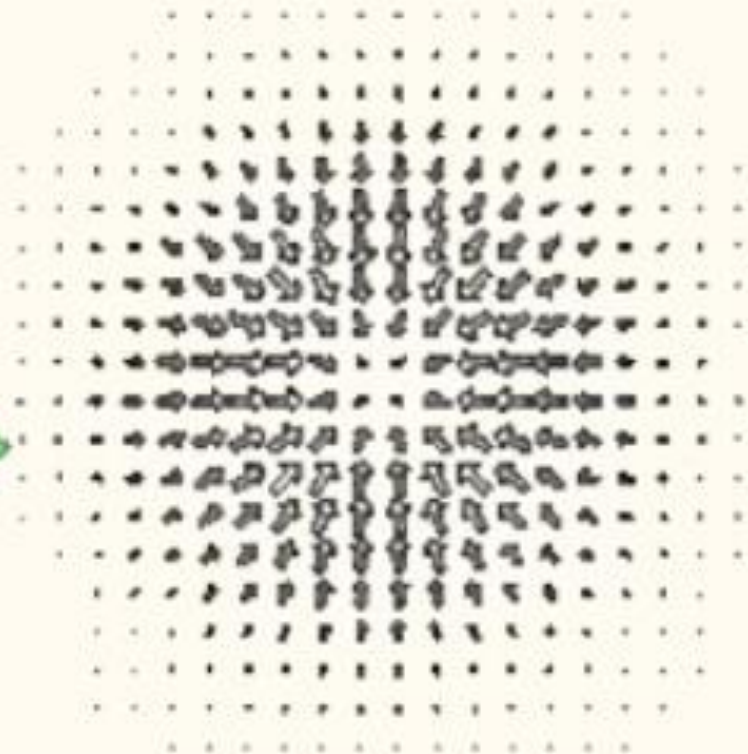
называется градиентом.

Линией уровня функции называется множество точек, в которых функция принимает одно и то же фиксированное значение. Оказывается, что градиент перпендикулярен линии уровня. Более подробное обсуждение этого факта будет произведено позднее.

Градиент

$\nabla\varphi$ $\quad$ $\operatorname{grad}\varphi$





Градиент в задачах оптимизации

Задачей оптимизации называется задача по нахождению экстремума функции, например минимума:

$$f(x_1, ..., x_n) \to \min .$$

Такая задача часто встречается в приложениях, например при выборе оптимальных параметров рекламной компании, а также в задачах классификации.

Если функция дифференцируема, то найти точки, подозрительные на экстремум, можно с помощью необходимого условия экстремума: все частные производные должны равняться нулю, а значит вектор градиента — нулевому вектору.

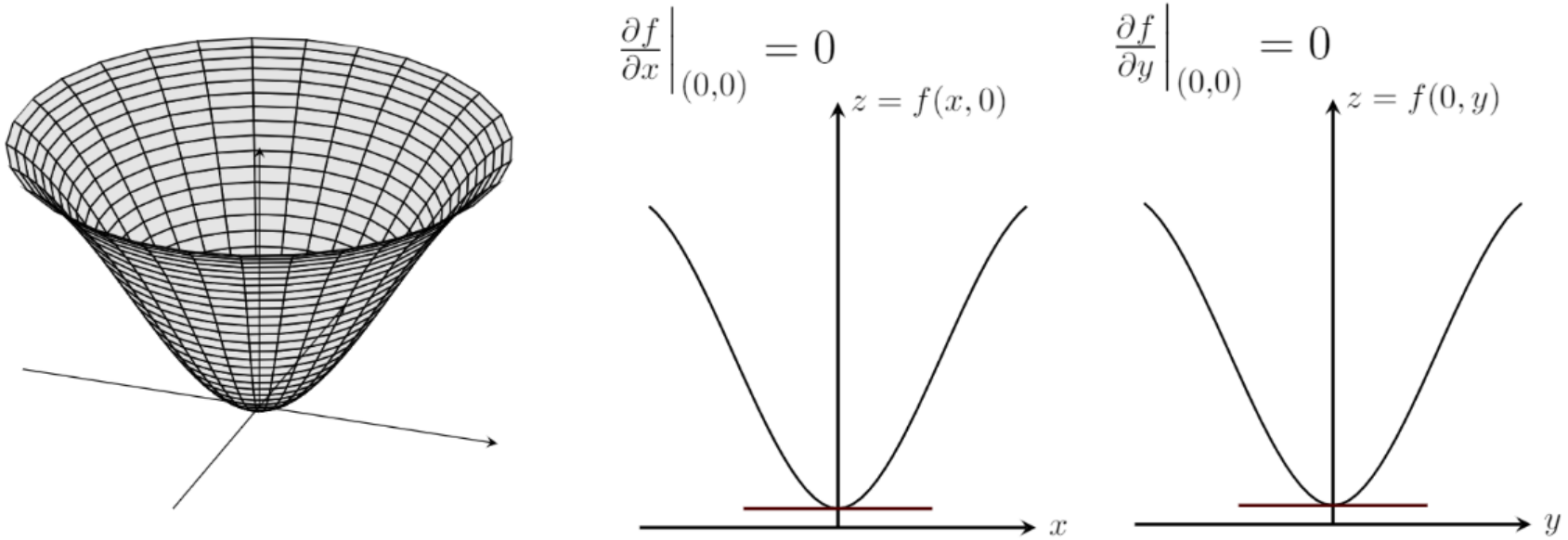
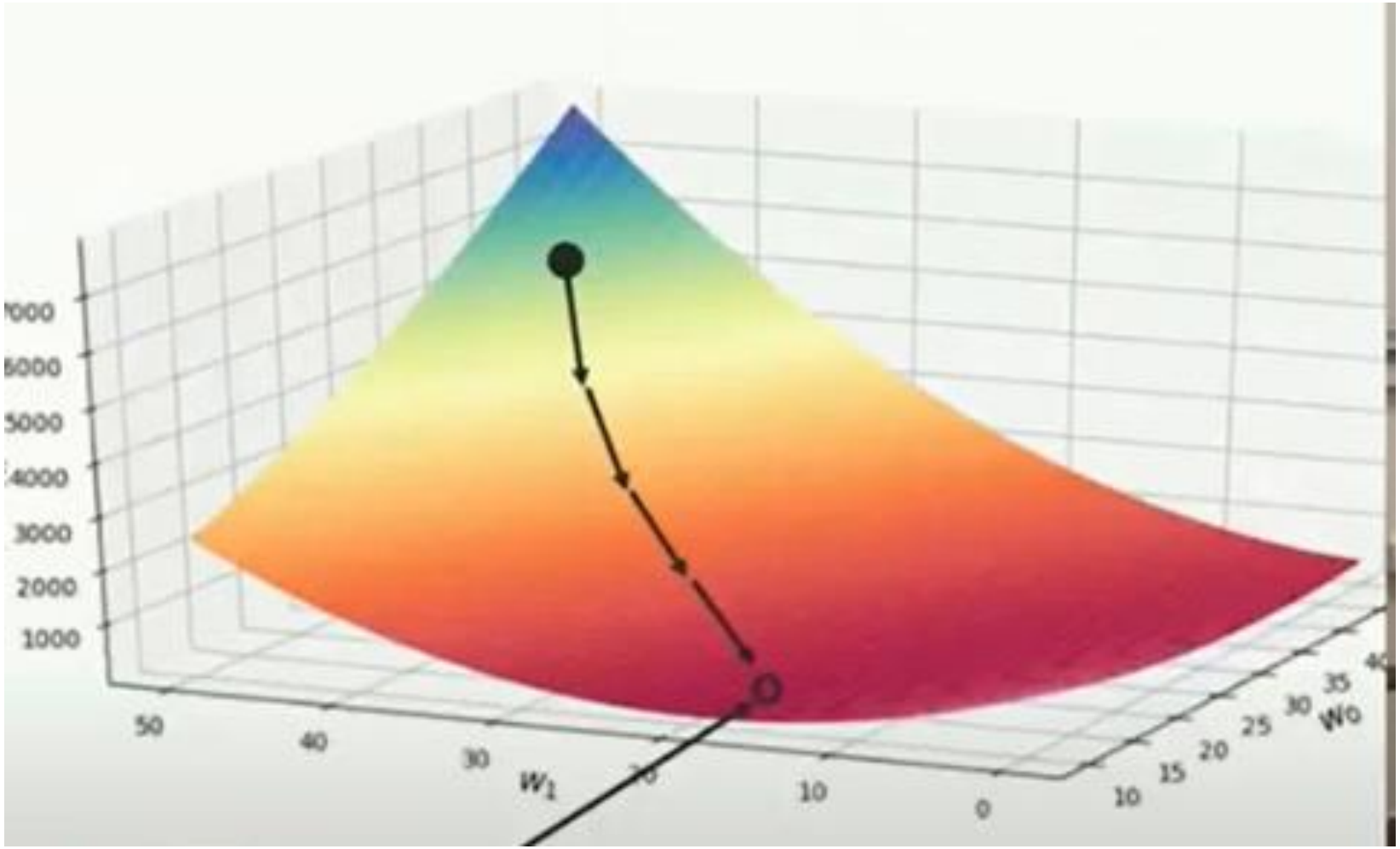


Рис. 2: Функция двух переменных достигает минимума в начале координат.


7000
6000
5000
4000
3000
2000
1000
50
40
30
w_1
10
0
10
15
20
25
30
35
w_0

Но не всегда задачу можно решать аналитически. В таком случае используется численная оптимизация.

Наиболее простым в реализации из всех методов численной оптимизации является метод градиентного спуска. Это итерационный метод. Решение задачи начинается с выбора начального приближения $\vec{x}^{[0]}$. После вычисляется приблизительное значение $\vec{x}^1$, а затем $\vec{x}^2$ и так далее, согласно итерационной формуле:

$$\vec{x}^{[j+1]} = \vec{x}^{[j]} - \gamma^{[j]} \nabla F(\vec{x}^{[j]}), \qquad \text{где } \gamma^{[j]} \text{ — шаг градиентного спуска.}$$

Основная идея метода заключается в том, чтобы идти в направлении наискорейшего спуска, а это направление задаётся антиградиентом $-\nabla F$.

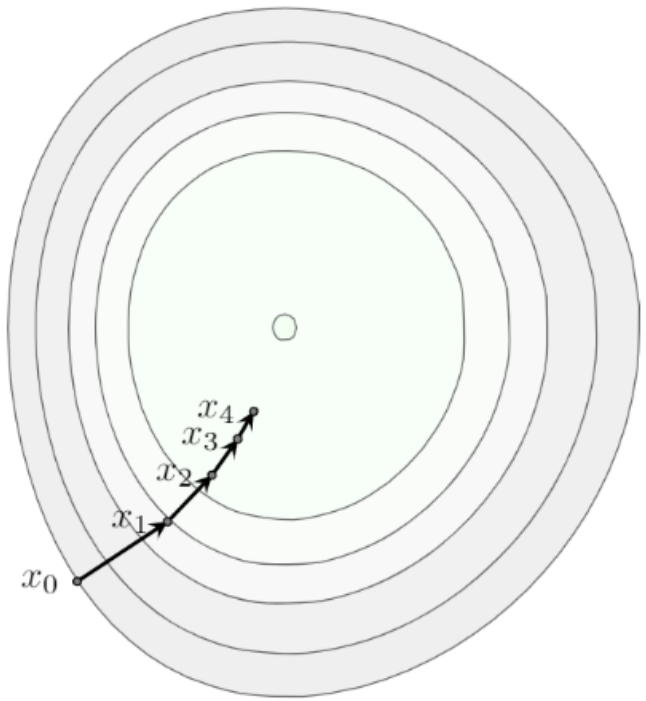
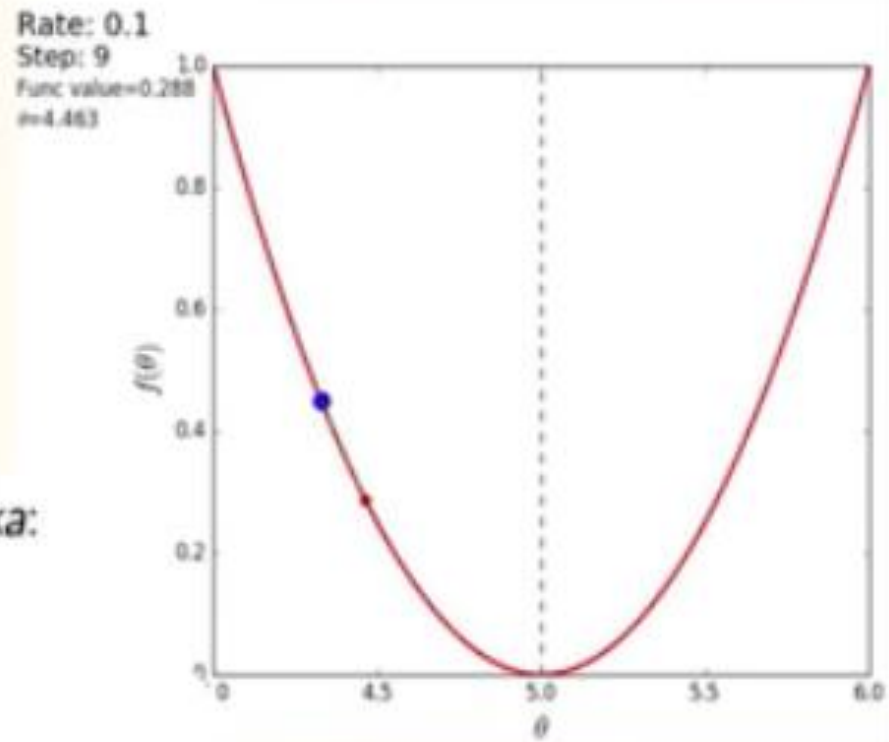


Рис. 3: Градиентный спуск

Градиентный спуск (GD)

$$Q(w) = \sum_{i=1}^{\ell} \mathscr{L}_i(w) \to \min_w$$



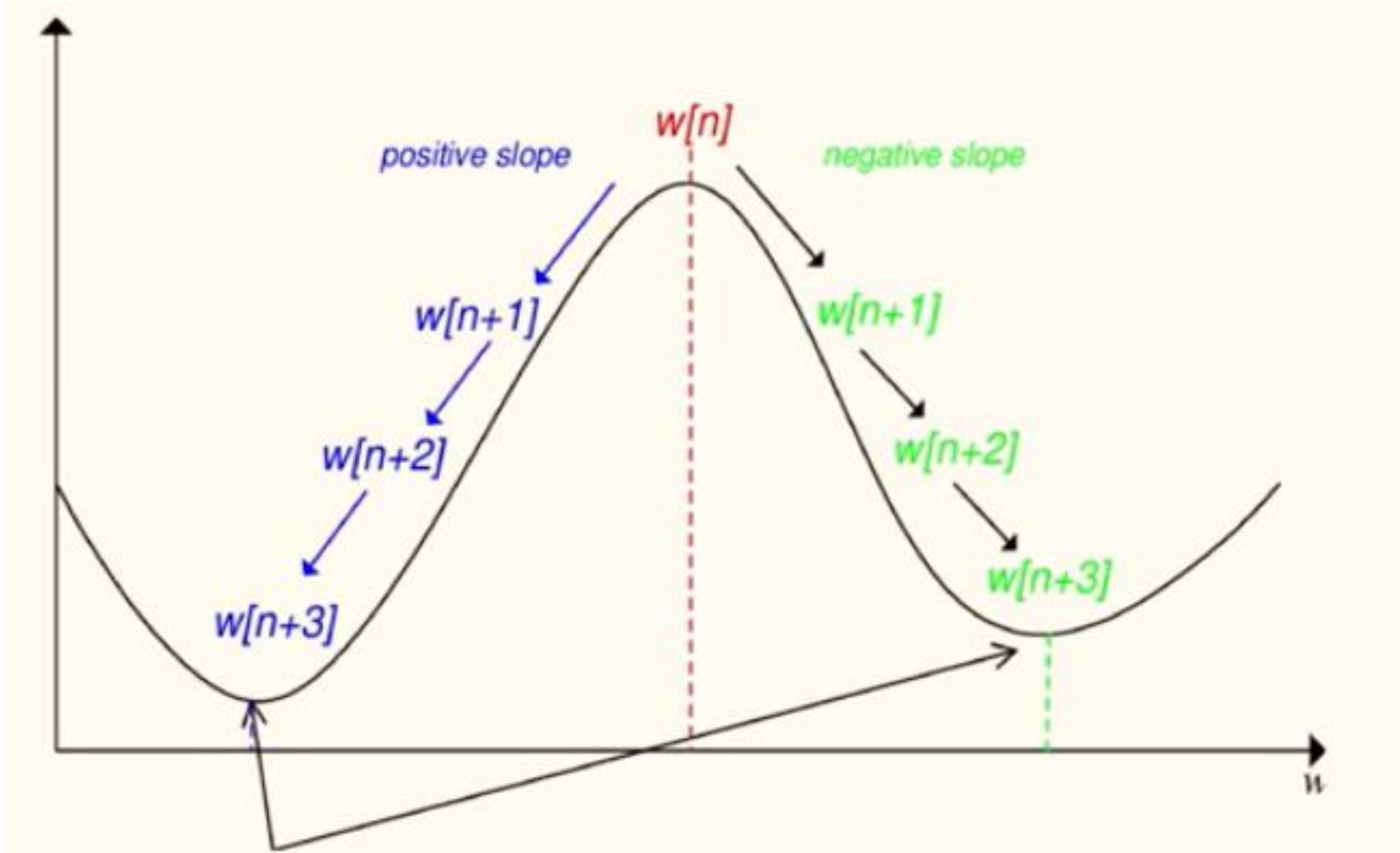
Численная минимизация методом *градиентного спуска*:

$w^{(0)} :=$ начальное приближение;

$$w^{(t+1)} := w^{(t)} - h \cdot \nabla Q(w^{(t)}),$$

где h — *градиентный шаг*, называемый также *темпом обучения*

$$w^{(t+1)} := w^{(t)} - h \sum_{i=1}^{\ell} \nabla \mathscr{L}_i(w^{(t)}).$$


w[n]
positive slope
negative slope
w[n+1]
w[n+2]
w[n+3]
w[n+1]
w[n+2]
w[n+3]
w

Направление наискорейшего спуска

Пусть $f(\vec{x}) = f(x_1, x_2, \ldots, x_n)$ — функция n переменных, $\vec{\ell} \in \mathbb{R}^n$, $|\vec{\ell}| = 1$, тогда частной производной в точке x_0 по направлению $\vec{\ell}$ называется

$$\frac{\partial f}{\partial \vec{\ell}}(\vec{x}_0) = \lim_{t \to 0} \frac{f(\vec{x}_0 + t \cdot \vec{\ell}) - f(\vec{x}_0)}{t}.$$

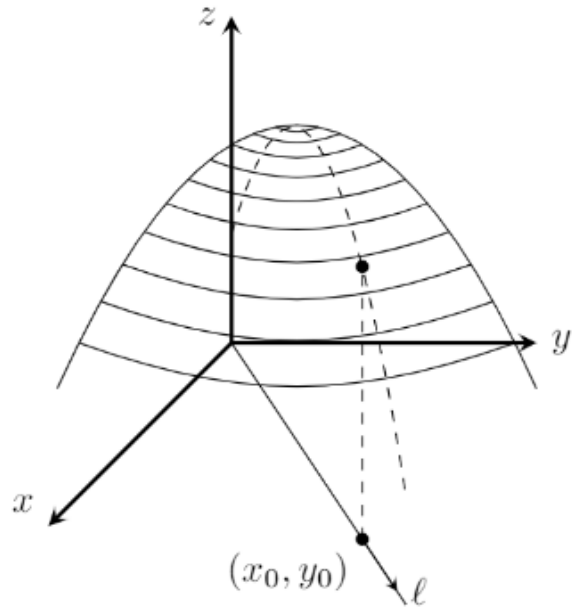


Рис. 4: К определению производной по направлению функции двух аргументов.

Непосредственно из определения становится понятен геометрический смысл производной по направлению. Производная по направлению показывает, насколько быстро функция изменяется при движении вдоль заданного направления. Производная по направлению координатной оси является частной производной по соответствующей координате.

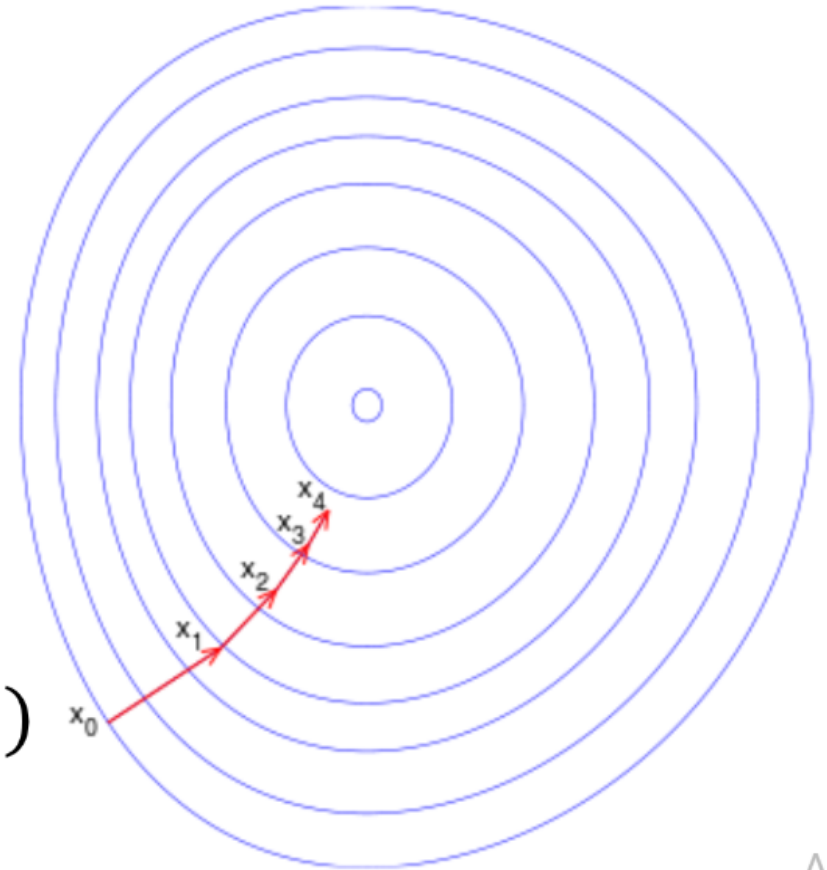
Problems

- Будет ли сходимость?
- Как выбрать начальное приближение?
- Как выбрать шаг ?
- Как ускорить?

Напоминание: градиентный спуск

$$\nabla F(x_k) = \begin{pmatrix} \frac{\partial F}{\partial x_{k1}} \\ \vdots \\ \frac{\partial F}{\partial x_{kn}} \end{pmatrix}$$

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



Проблема оптимизации параметров алгоритма машинного обучения

На самом деле, использовать методы оптимизации функций, требующие существование градиента, получается не всегда. Даже если градиент у интересующей функции существует, часто оказывается, что вычислять его непрактично.

Например, существует проблема оптимизации параметров $\alpha_1, ..., \alpha_N$ некоторого алгоритма машинного обучения. Задача оптимизации состоит в том, чтобы подобрать эти параметры так, чтобы алгоритм давал наилучший результат. В частности, если качество работы алгоритма описывать функцией качества $Q(\alpha_1, ..., \alpha_N)$ от его параметров, задача оптимизации принимает вид:

$$Q(\alpha_1, ..., \alpha_N) \to \max_{\alpha_1, ..., \alpha_N}.$$

Вычисление градиента в этом случае или невозможно в принципе, или крайне непрактично.

Проблема локальных минимумов

Другая проблема градиентных методов — проблема локальных минимумов.

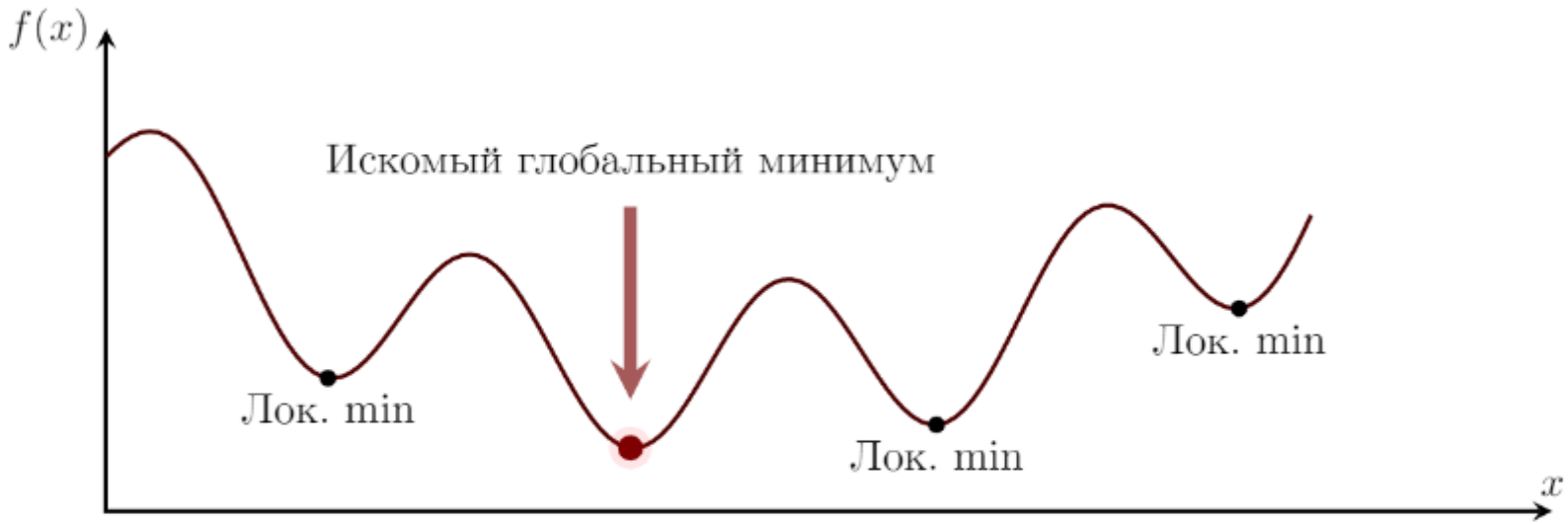


Рис. 1: К вопросу о проблеме локальных минимумов

Метод градиентного спуска, попав на дно локального минимума, где градиент также равен нулю, там и остается. Глобальный минимум так и остается не найденным. Решить эти проблемы позволяют методы случайного поиска. Общая идея этих методов заключается в намеренном введении элемента случайности.

Ускорения сходимости к минимуму?

1. Брать по одной новой паре (x,y) и сразу обновлять вектор весов
2. Просматривать не в одном порядке, а в случайном

$$\mathbf{GD + 1) + 2) = SGD}$$

(стохастический градиентный спуск)

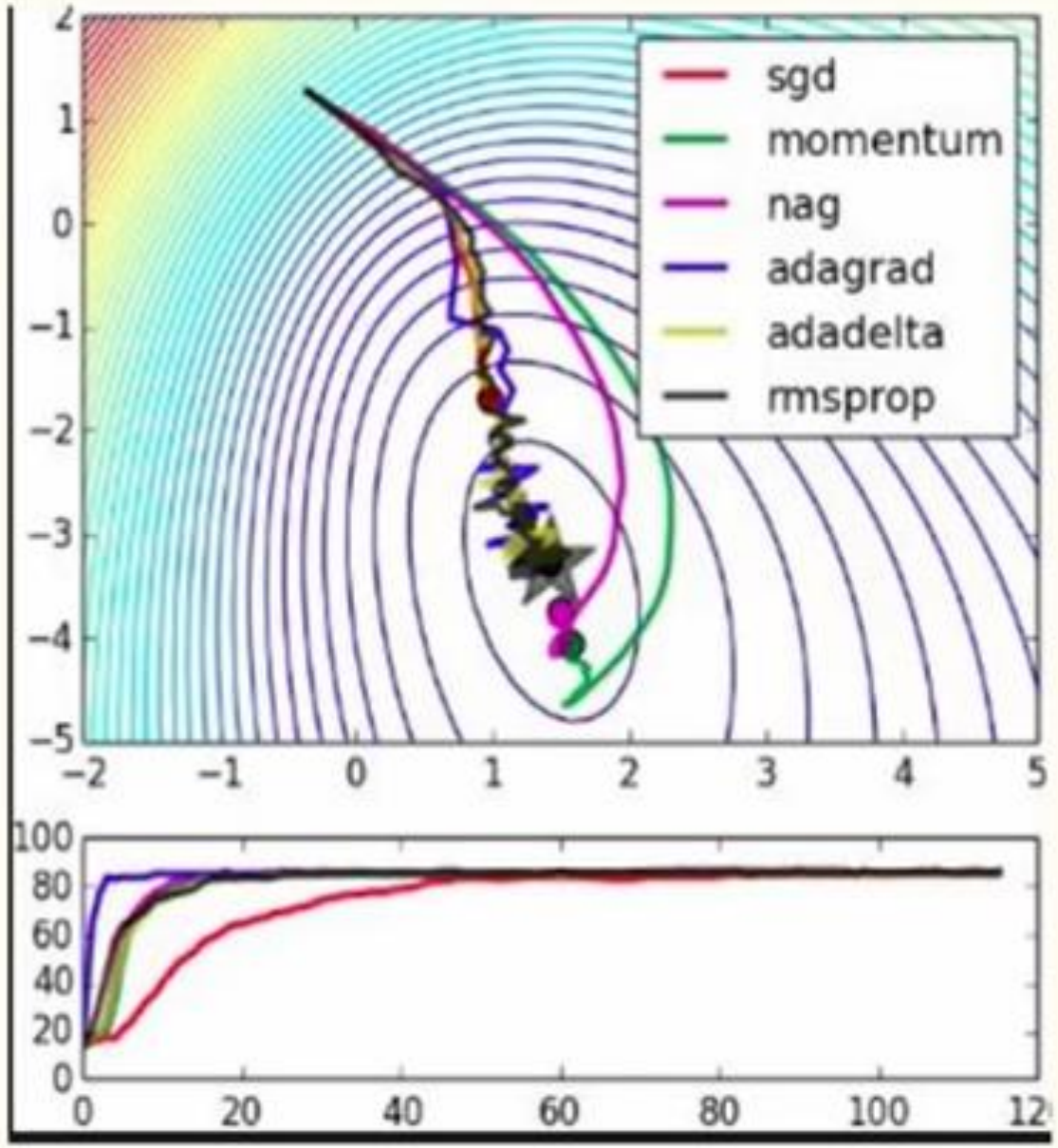
Пример: градиентный спуск без градиента

Если градиент по каким-либо причинам вычислить нельзя, задачу оптимизации можно попытаться решить с помощью модифицированного стохастического алгоритма градиентного спуска.

Пусть решается задача оптимизации $f(\vec{x}) \to \min_{\vec{x}}$, зафиксировано некоторое число d — параметр метода. Используется следующий итерационный процесс:

1. Случайным образом выбирается вектор $\vec{u}$ (случайный вектор $\vec{u}$ равномерно распределен по сфере).
2. Вычисляется значение выражения, которое есть ни что иное, как численная оценка значения производной функции f по направлению $\vec{u}$:
$$\frac{f(\vec{x}) - f(\vec{x} + d\vec{u})}{d}.$$
3. Сдвигаем точку в направлении $\vec{u}$ пропорционально вычисленной на предыдущем шаге величине.

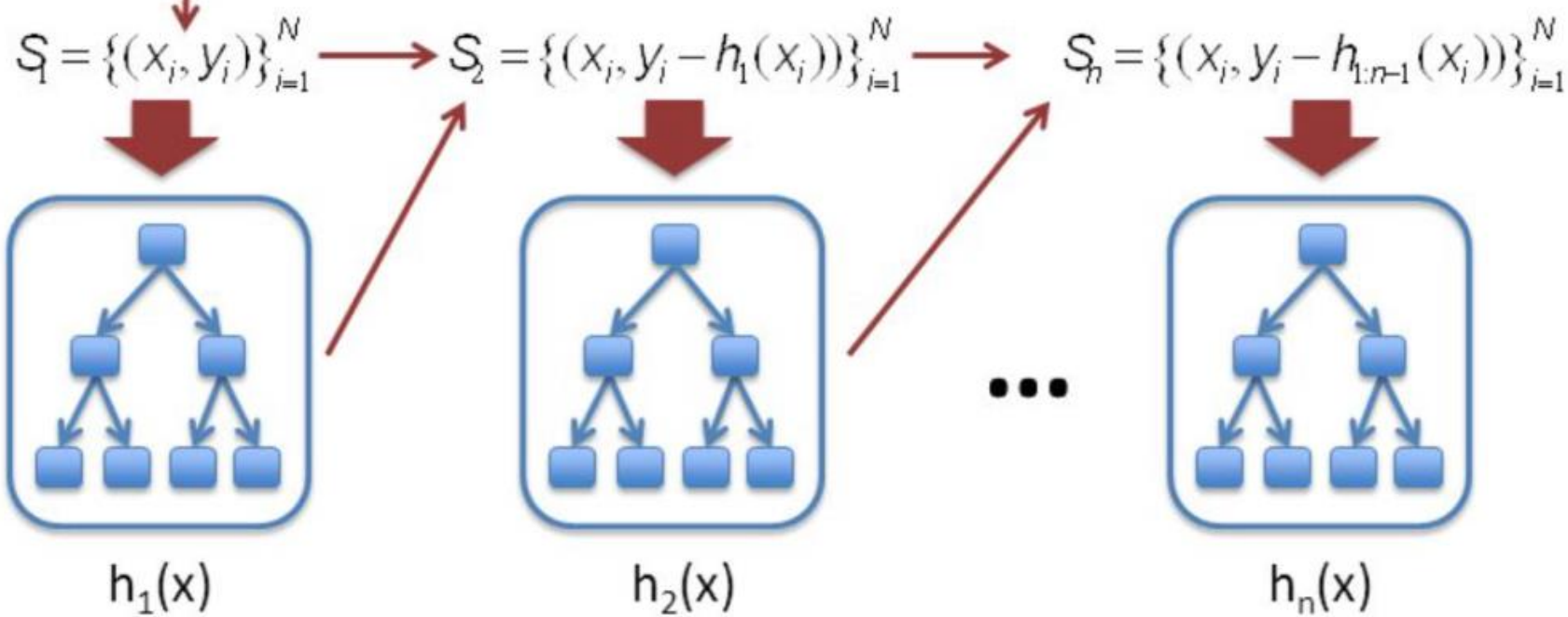
Следует отметить, что ни на каком шаге градиент функции не вычисляется. Более того, направление смещения $\vec{u}$ выбирается случайным образом. Но так как величина смещения зависит от выражения функции в точке $\vec{x} + d\vec{u}$, в среднем происходит смещение по антиградиенту сглаженной функции. Причем, зафиксированное в начале число d, как раз имеет смысл «параметра сглаживания» при нахождении численной оценки производной.


sgd
momentum
nag
adagrad
adadelta
rmsprop
2
1
0
−1
−2
−3
−4
−5
−2
−1
0
1
2
3
4
5
100
80
60
40
20
0
0
20
40
60
80
100

Идея Gradient Boosted Decision Trees (GBDT)

$S = \{(x_i, y_i)\}_{i=1}^{N}$

$h(x) = h_1(x) + h_2(x) + \ldots + h_n(x)$



Где здесь градиент

$$L\big(y_i, h(x_i)\big) = (y_i - h(x_i))^2$$

$$Q = \sum_{i=1}^{N} L\big(y_i, h(x_i)\big) = \sum_{i=1}^{N} (y_i - h(x_i))^2$$

Где здесь градиент

$$L\big(y_i, h(x_i)\big) = (y_i - h(x_i))^2$$

$$Q = \sum_{i=1}^{N} L\big(y_i, h(x_i)\big) = \sum_{i=1}^{N} (y_i - h(x_i))^2$$

$$\frac{\partial Q}{\partial h_i} = \sum_{i=1}^{N} \frac{\partial}{\partial h_i} (y_i - h_i)^2 = 2(y_i - h_i)$$

Где здесь градиент

$$\frac{\partial Q}{\partial h_i} = \sum_{i=1}^{N} \frac{\partial}{\partial h_i} (y_i - h_i)^2 = 2(y_i - h_i)$$

$$h(x_i) = \sum_{k=1}^{n} \alpha_k h_k(x_i)$$

Новый $h_k(x)$ будем обучать на ответы $y_i - h_i$

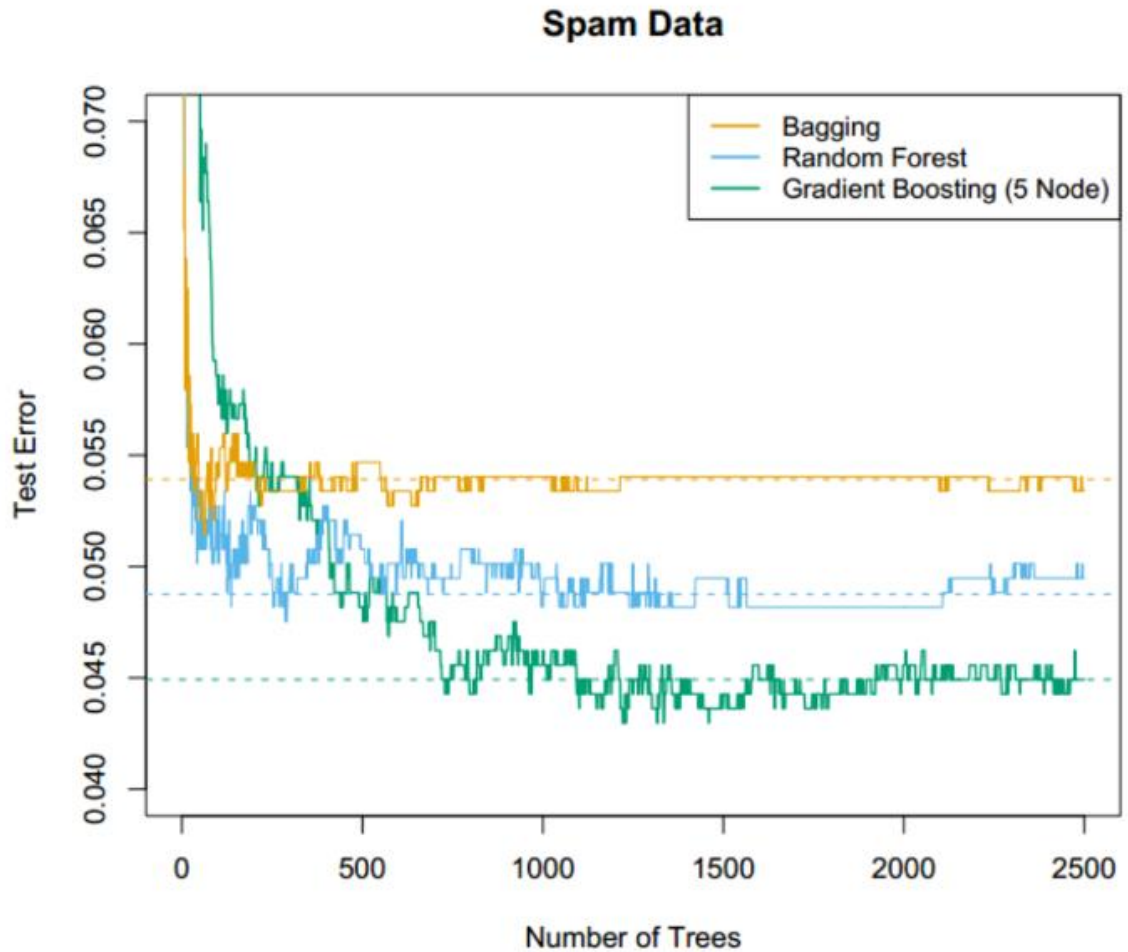
Gradient Boosted Decision Trees

- Каждое новое дерево $h_k(x)$ обучаем на ответы $y_i - h_i$

  h_i - прогноз всей композиции на i-том объекте на предыдущей итерации

- Коэффициент α_k перед новым деревом подбираем с помощью численной оптимизации ошибки Q

GBDT и RF


Spam Data
Bagging
Random Forest
Gradient Boosting (5 Node)
Test Error
0.040
0.045
0.050
0.055
0.060
0.065
0.070
0
500
1000
1500
2000
2500
Number of Trees

Библиотеки

- Scikit-learn:
  - sklearn.ensemble.RandomForestClassifier
  - sklearn.ensemble.RandomForestRegressor
- XGBoost