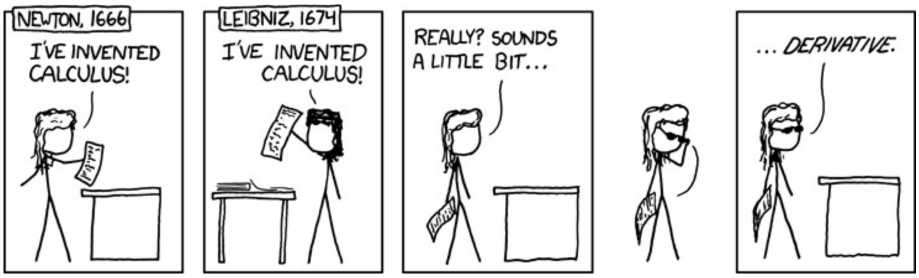
Дифференциальное исчисление и градиентный спуск



Сложные и важные части отмечены оранжевым в тексте конспекта или в выносках

Некоторые обозначения

$\mathbb{R}$ – множество действительных чисел

$U(x_0)$ – окрестность точки x_0,

$f \in C^r(\Omega)$ – функция *f* определена на множестве Ω и имеет непрерывные производные до r-й включительно.

Понятие производной (derrivative)

Пусть в некоторой окрестности точки $x_0 \in \mathbb{R}$ определена функция

$$f : U(x_0) \subset \mathbb{R} \to \mathbb{R}$$

функция определённая на множестве действительных чисел, с областью определения в области действительных чисел

Производной функции f в точке x_0 называется предел (если он существует):

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

Общепринятые обозначения производной функции $y = f(x)$ в точке x_0

$$f'(x_0) = f'_x(x_0) = \mathrm{D}f(x_0) = \frac{df}{dx}(x_0) = \left.\frac{dy}{dx}\right|_{x=x_0} = \dot{y}(x_0)$$

Последнее обозначение $\dot{y}$ обычно обозначает производную по времени (в теоретической механике и физике, исторически часто тоже).

Производная $f'(x_0)$ функции *f* в точке x_0, будучи пределом, может не существовать или существовать и быть конечной или бесконечной. Функция *f* является дифференцируемой в точке x_0 тогда и только тогда, когда её производная в этой точке существует и конечна:

$$f \in \mathcal{D}(x_0) \Leftrightarrow \exists f'(x_0) \in (-\infty; \infty)$$

См. также разрывы функции в точке.

Дифференцируемую функцию *f* можно аппроксимировать в окрестности $U(x_0)$ следующей функцией:

см. также ряд Тейлора.

$f(x) = f(x_0) + f'(x_0)(x - x_0) + o(x - x_0)$ при $x \to x_0$

где $o(x - x_0)$ – «о малое от $x - x_0$» обозначает «бесконечно малое относительно $x - x_0$»

см. также уравнение прямой по двум точкам

Касательная к графику

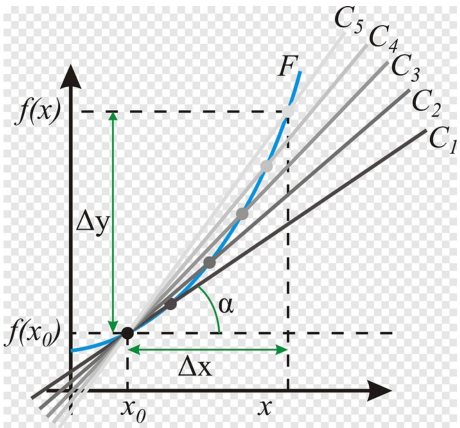
Пусть $f\colon U(x_0) \to \mathbb{R}$ и $x_1 \in U(x_0)$. Тогда прямая линия, проходящая через точки $(x0, f(x0))$ и $(x_1, f(x_1))$ задаётся уравнением

$$y = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0).y = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0).$$

Эта прямая проходит через точку $(x_0, f(x_0))(x_0, f(x_0))$ для любого $x_1 \in U(x_0)$, и её угол наклона $\alpha(x_1)$ удовлетворяет уравнению

$$\operatorname{tg}\,\alpha(x_1) = \frac{f(x_1) - f(x_0)}{x_1 - x_0}.$$

В силу существования производной функции f в точке x_0, переходя к пределу при $x_1 \to x_0$, получаем, что существует предел $\lim\limits_{x_1 \to x_0} \operatorname{tg}\,\alpha(x_1) = f'(x_0)$, а в силу непрерывности арктангенса и предельный угол $\alpha = \operatorname{arctg}\, f'(x_0)$.



Секущая переходит в касательную при уменьшении приращения x

Прямая, проходящая через точку $(x_0, f(x_0))$ и имеющая предельный угол наклона, удовлетворяющий $\operatorname{tg}\,\alpha = f'(x_0)$, задаётся уравнением касательной:

$$y = f(x_0) + f'(x_0)(x - x_0)$$

Таблица производных

ru.wikipedia.org/wiki/Таблица_производных

Правила дифференцирования

Если C – постоянное число, $f = f(x)$, $g = g(x)$– некоторые дифференцируемые функции, то справедливы следующие правила дифференцирования:

- $C' = 0$
- $x' = 1$
- $(f + g)' = f' + g'$
- $(fg)' = f'g + fg'$
- $(Cf)' = Cf'$

- $\left(\frac{f}{g}\right)' = \frac{f'g - fg'}{g^2}$ $\left(\frac{f}{g}\right)' = \frac{f'g - fg'}{g^2}$ для (g ≠ 0)

- Если функция задана параметрически:

$$\begin{cases} x = x(t), \\ y = y(t), \end{cases} t \in [T_1; T_2] \begin{cases} x = x(t), \\ y = y(t), \end{cases} t \in [T_1; T_2]$$

то

$$y'_x = \frac{dy}{dx} = \frac{dy}{dt} \cdot \frac{dt}{dx} = y'_t \cdot t'_x = \frac{y'_t}{x'_t} y'_x = \frac{dy}{dx} = \frac{dy}{dt} \cdot \frac{dt}{dx} = y'_t \cdot t'_x = \frac{y'_t}{x'_t}$$

- Дифференцирование сложной функции – chain rule (см. примеры 1 и 2 ниже) [!!!]

$$\frac{d}{dx} f(g(x)) = \frac{df(g)}{dg} \cdot \frac{dg(x)}{dx} = f'_g g'_x \frac{d}{dx} f(g(x)) = \frac{df(g)}{dg} \cdot \frac{dg(x)}{dx} = f'_g g'_x$$

- Формулы производной произведения и отношения обобщаются на случай n-кратного дифференцирования (формула Лейбница):

$(fg)^{(n)} = \sum_{k=0}^{n} C_n^k f^{(n-k)} g^{(k)}, (fg)^{(n)} = \sum_{k=0}^{n} C_n^k f^{(n-k)} g^{(k)}$, где C_n^k – биномиальные коэффициенты (количество сочетаний из n по k)

$$\binom{n}{k} = C_n^k = \frac{n!}{k!\,(n-k)!}$$

Примеры

Пример 1. Производная сложной функции

$$y = \underbrace{\cos \underbrace{2x}_{g(x)}}_{f(g(x))}$$

сложная функция

$$y' = f'_g \cdot g'_x \; ; \qquad g'_x = (2x)' = 2$$

$$f'_g = -\sin 2x$$

$$y' = -2\sin 2x$$

Пример 2. Производная сложной функции

$$y = \sqrt{3 - \cos^3(\ln[x + \sqrt{x}])}$$

$$y' = \frac{1}{2\sqrt{3-\cos^3(\ln[x+\sqrt{x}])}} \cdot \left(-\cos^3(\ln[x+\sqrt{x}])\right)' =$$

$$= \frac{1}{2\sqrt{3-\cos^3(\ln[x+\sqrt{x}])}} \cdot \left(-3\cos^2(\ln[x+\sqrt{x}])\right) \cdot$$

$$\cdot \left(\cos(\ln[x+\sqrt{x}])\right)' =$$

$$= \frac{+1}{2\sqrt{\ldots}} \cdot 3\cos^2(\ln[x+\sqrt{x}]) \cdot \sin(\ln[x+\sqrt{x}]) \cdot \left(\ln[x+\sqrt{x}]\right)' =$$

$$= \frac{3\cos^2(\ln[x+\sqrt{x}]) \cdot \sin(\ln[x+\sqrt{x}])}{2\sqrt{3-\cos^3(\ln[x+\sqrt{x}])}} \cdot \frac{(x+\sqrt{x})'}{x+\sqrt{x}} =$$

$$= \frac{3\cos^2(\ln[x+\sqrt{x}]) \cdot \sin(\ln[x+\sqrt{x}]) \cdot}{2 \cdot \sqrt{3-\cos^3(\ln[x+\sqrt{x}])} \cdot (x+\sqrt{x})} \cdot \left(1 + \frac{1}{2\sqrt{x}}\right)$$

Пример 3. Иногда стоит функцию упростить перед взятием производной

$$y = \ln\sqrt[5]{\frac{1-x}{1+x}}$$

Упростим выражение:

– степень, под логарифмом, можно вынести: $\ln a^b = b\ln a$

– логарифм представить на разность $\ln(a/b) = \ln a - \ln b$

$$y = \ln\sqrt[5]{\frac{1-x}{1+x}} = \frac{1}{5}\ln\left(\frac{1-x}{1+x}\right) =$$

$$= \frac{1}{5}\left[\ln(1-x) - \ln(1+x)\right] =$$

$$y' = \frac{1}{5}\left[\frac{-1}{1-x} - \frac{+1}{1+x}\right]$$

Пример 4. Вычислить приблизительное значение функции в точках, если известно её значение и значение её производной в некоторой, близкой к заданным, точке. Сравнить эти значения с точным значением функции.

Используем разложение функции в ряд Тейлора, ограничившись двумя членами ряда.

см. выше формулу для аппроксимацию функции в окрестности точки

$y = \sin(x)$

$x_0 = \pi/4$, $\sin\frac{\pi}{4} \approx 0.71$

$x_1 = 0.9$ рад. $\cos\frac{\pi}{4} \approx 0.71$

$x_2 = 1$ рад.

$$y \approx y\left(\frac{\pi}{4}\right) + y'\left(\frac{\pi}{4}\right)\cdot\left(x - \frac{\pi}{4}\right)$$

$$y \approx 0.71 + 0.71\left(x - \frac{\pi}{4}\right)$$

$$y(0.9) \approx 0.71 + 0.71\cdot 0.11 \approx 0.79$$

$$\sin(0.9) = 0.78$$

$$y(1) \approx 0.71 + 0.71\cdot 0.21 \approx 0.86$$

$$\sin(1) \approx 0.84$$

Производная и экстремумы функции

Критической точкой дифференцируемой функции $f : \mathbb{R}^n \to \mathbb{R}$ называется точка, в которой её дифференциал обращается в нуль. Это условие эквивалентно тому, что в данной точке все частные производные первого порядка обращаются в нуль, геометрически оно означает, что касательная гиперплоскость к графику функции горизонтальна.

Экстремум — максимальное или минимальное значение функции на заданном множестве.

Точкой экстремума может быть только критическая точка. Но не все критические точки – точки экстремума. Пример – точка перегиба в кубической параболе.

Теорема Ферма: производная дифференцируемой функции в точке локального экстремума равна нулю.

Доказательство.

Предположим, что $f(x_0) = \max\limits_{x \in (a,b)} f(x)$. Тогда $\forall x \in (a,b)\colon\ f(x) \leqslant f(x_0)$.

Поэтому:

$$f'_-(x_0) = \lim_{x \to x_0 - 0} \left(\frac{f(x) - f(x_0)}{x - x_0} \right) \geqslant 0,$$

Что значит x0±0 в примере?

$$f'_+(x_0) = \lim_{x \to x_0 + 0} \left(\frac{f(x) - f(x_0)}{x - x_0} \right) \leqslant 0.$$

Если производная $f'(x_0)$ определена, то получаем $0 \leqslant f'_-(x_0) = f'(x_0) = f'_+(x_0) \leqslant 0$, то есть $f'(x_0) = 0$

Если x_0 — точка локального минимума функции f то доказательство аналогично.

Достаточные условия существования локальных экстремумов

Пусть дана функция $f : M \subset \mathbb{R} \to \mathbb{R}$ и $x_0 \in M^0$ — внутренняя точка области определения *f*. Тогда:

- x_0 называется точкой локального максимума функции f, если существует проколотая окрестность $\dot{U}(x_0)$ такая, что $\forall x \in \dot{U}(x_0) \rightarrow \ f(x) \leqslant f(x_0)$
- x_0 называется точкой локального минимума функции f, если существует проколотая окрестность $\dot{U}(x_0)$ такая, что $\forall x \in \dot{U}(x_0) \rightarrow f(x) \geqslant f(x_0)$
- x_0 называется точкой глобального (абсолютного) максимума, если $\forall x \in M \rightarrow f(x) \leqslant f(x_0)$
- x_0 называется точкой глобального (абсолютного) минимума, если $\forall x \in M \rightarrow \ f(x) \geqslant f(x_0)$

Достаточное условие экстремума: если в точке *x* производная $f'(x)$ функции $f(x)$ меняет знак, то точка *x* является экстремумом. Если производная становится

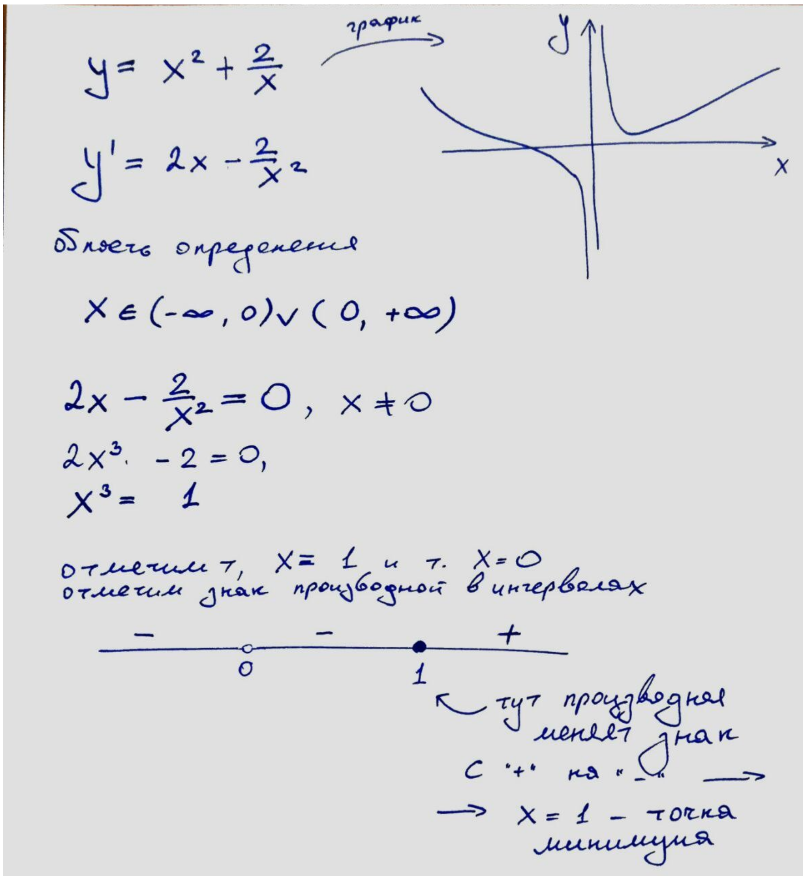
положительной, то то точка минимума, в противном случае точка максимума.

См. также

ru.wikipedia.org/wiki/Численное_дифференцирование

wolframalpha.com/examples/mathematics/calculus-and-analysis/derivatives – вычисление производных в WolframAlpha

Пример 5. Поиск экстремумов



Если функция задана уравнением y = f (x), разрешённым относительно y, то функция задана в явном виде (явная функция).

Под неявным заданием функции понимают задание функции в виде уравнения F (x; y)= 0, не разрешённого относительно y.

Всякую явно заданную функцию y = f (x) можно записать как неявно заданную уравнением f (x) - y = 0, но не наоборот.

Пример. Производная функции заданной неявно [!!!]

это не производная функции нескольких переменных

$$3x^2y^2 - 5x + \sin y = 3y - 1$$

неявно заданная функция

Найдём производную.

— будем брать производную и от x и от y

— при взятии производной от y нужно помнить, что это производная сложной функции

$$3(x^2y^2)' - 5 + (\sin y)' = 3y'$$

производн. произведения, производн. сложн. функ.

производная сложной функции

$$6xy^2 + 6x^2yy' - 5 + \cos y \cdot y' = 3y'$$

производные от y влево:

$$6x^2yy' + \cos y \cdot y' = -6xy^2 + 5$$

$$y'(6x^2y + \cos y) = 5 - 6xy^2$$

$$y' = \frac{5 - 6xy^2}{6x^2y + \cos y}$$

Частная производная (partial derivative)

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

Частная производная функции $f(x_1, x_2, \ldots, x_n)$ в точке $(a_1, a_2, \ldots, a_n)$ определяется так:

$$\frac{\partial f}{\partial x_k}(a_1, \cdots, a_n) = \lim_{\Delta x \to 0} \frac{f(a_1, \ldots, a_k + \Delta x, \ldots, a_n) - f(a_1, \ldots, a_k, \ldots, a_n)}{\Delta x}$$

Обозначение $\frac{\partial f}{\partial x} \neq \frac{df}{dx}$

Производная по направлению

Производная по направлению — одно из обобщений понятия производной на случай функции нескольких переменных. Производная по направлению показывает, как быстро значение функции изменяется при движении в данном направлении.

Рассмотрим дифференцируемую функцию $f(x_1, \ \ldots, \ x_n)$ от n аргументов в окрестности точки $\vec{x}^{\,0} = (x_1^0, \ \ldots, \ x_n^0)$. Для любого единичного вектора $\vec{e} = (e_1, \ \ldots, \ e_n)$ определим производную функции f в точке $\vec{x}^{\,0}$ по направлению $\vec{e}$:

∇ — оператор набла

$$\nabla_{\mathbf{e}} f(\mathbf{x}) = \lim_{h \to 0} \frac{f(\vec{x}^{\,0} + h \cdot \vec{e}) - f(\vec{x}^{\,0})}{h}$$

Значение этого выражения показывает, как быстро меняется значение функции при сдвиге аргумента в направлении вектора $\vec{e}$.

Если направление сонаправленно с координатной осью, то производная по этому направлению совпадает с частной производной по этой координате.

Производная по направлению и частные производные

$$\nabla_{\mathbf{e}} f(\mathbf{x}) = \frac{\partial f}{\partial x_1} e_1 + \cdots + \frac{\partial f}{\partial x_n} e_n$$

Формулу можно записать короче, через скалярное произведение:

$$\nabla_{\mathbf{e}} f(\mathbf{x}) = \nabla f \cdot \vec{e}$$

$$\vec{x} \cdot \vec{y} = \sum_{i=1}^{n} x_i \cdot y_i$$ *← скалярное произведение*

Градиент (Gradient)

Для случая трёхмерного пространства градиентом скалярной функции $\varphi = \varphi(x, y, z)$ координат x, y, z называется векторная функция с компонентами

$$\frac{\partial\varphi}{\partial x}, \frac{\partial\varphi}{\partial y}, \frac{\partial\varphi}{\partial z}.$$

Или, использовав для единичных векторов по осям прямоугольных декартовых координат $\vec{e}_x$,$\vec{e}_y$,$\vec{e}_z$:

$$\operatorname{grad}\varphi = \nabla\varphi = \frac{\partial\varphi}{\partial x}\vec{e}_x + \frac{\partial\varphi}{\partial y}\vec{e}_y + \frac{\partial\varphi}{\partial z}\vec{e}_z$$

Если φ — функция n переменных $x_1, \ \ldots, \ x_n x_1, \ \ldots, \ x_n$, то её градиентом называется n-мерный вектор

$$\left(\frac{\partial\varphi}{\partial x_1}, \ \ldots, \ \frac{\partial\varphi}{\partial x_n}\right)$$

компоненты которого равны частным производным φ по всем её аргументам.

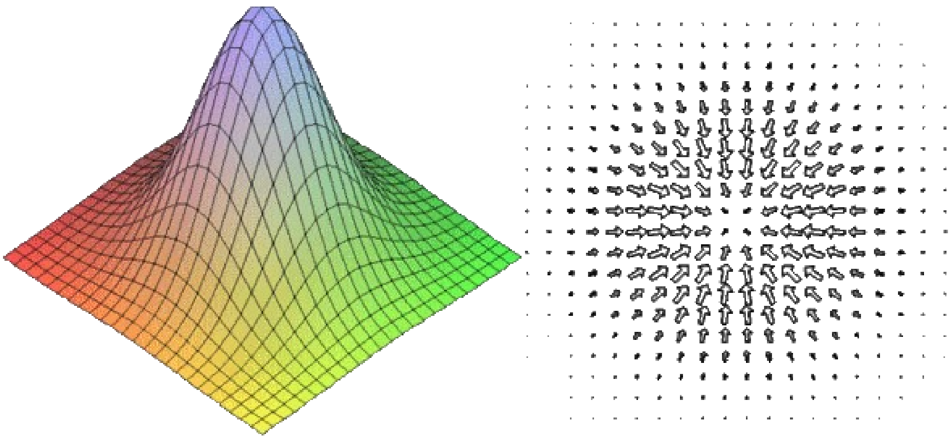


График функции (слева) и вектора её градиента (справа)

Свойства градиента

- grad c = 0, где c – константа
- Свойства линейности:
  - $grad(c \cdot \vec{u}) = c \cdot grad(\vec{u})$
  - $grad(\vec{u} + \vec{v}) = grad(\vec{u}) + grad(\vec{v})$
  - $grad(\vec{u} \cdot \vec{v}) = \vec{u} \cdot grad(\vec{v}) + \vec{v} \cdot grad(\vec{u})$

см. правила дифференцирования выше

Матрица Якоби – матрица производных

Пусть задано отображение $\mathbf{u} : \mathbb{R}^n \to \mathbb{R}^m$, $\mathbf{u} = (u_1, \ldots, u_m)$, $u_i = u_i(x_1, \ldots, x_n), i = 1, \ldots, m,$ имеющее в некоторой точке *x* все частные производные первого порядка. Матрица J, составленная из частных производных этих функций в точке *x*, называется матрицей Якоби данной системы функций.

$$J(x) = \begin{pmatrix} \frac{\partial u_1}{\partial x_1}(x) & \frac{\partial u_1}{\partial x_2}(x) & \cdots & \frac{\partial u_1}{\partial x_n}(x) \\ \frac{\partial u_2}{\partial x_1}(x) & \frac{\partial u_2}{\partial x_2}(x) & \cdots & \frac{\partial u_2}{\partial x_n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial u_m}{\partial x_1}(x) & \frac{\partial u_m}{\partial x_2}(x) & \cdots & \frac{\partial u_m}{\partial x_n}(x) \end{pmatrix}$$

Иными словами, матрица Якоби является производной векторной функции от векторного аргумента.
Определитель этой матрицы называется якобианом.

Примеры

Пример 1. Найти градиент функции в точке

$u = x + \ln(z^2+y^2)$

$M_0(2, 1, 1)$

$\frac{\partial u}{\partial x} = 1$; $\frac{\partial u}{\partial y} = \frac{2y}{z^2+y^2}$; $\frac{\partial u}{\partial z} = \frac{2z}{z^2+y^2}$

$\left.\frac{\partial u}{\partial x}\right|_{M_0} = 1$ $\left.\frac{\partial u}{\partial y}\right|_{M_0} = 1$ $\left.\frac{\partial u}{\partial z}\right|_{M_0} = 1$

$\text{grad } u = (1, 1, 1)$

Пример 2. Найти производную функции в точке по направлению

$z = e^{x^2 y - x}$

$M_0(2, 0)$,

Найти произв. в т. M_0 в направлении, составляющем угол $\frac{\pi}{6}$ с положительным направлением оси OX

$\frac{\partial z}{\partial x} = (2xy - 1)e^{x^2 y - x}$; $\left.\frac{\partial z}{\partial x}\right|_{M_0} = -e^{-2}$

$\frac{\partial z}{\partial y} = x^2 \cdot e^{x^2 y - x}$; $\left.\frac{\partial z}{\partial y}\right|_{M_0} = 4e^{-2}$

$l = \left(\cos\frac{\pi}{6}, \sin\frac{\pi}{6}\right) \longleftarrow$ вектор направления

$\nabla_l z(M_0) = \left(-e^{-2}\cdot\cos\frac{\pi}{6}, 4e^{-2}\cdot\sin\frac{\pi}{6}\right) \approx$
$\approx (-0.12, 0.27)$

вектор направления должен быть единичным Т.е. его нужно нормировать при необходимости.

Пример 3

Найти минимум функции двух переменных.

Описать итерации метода градиентного спуска до тех пор, пока изменение значение минимизируемой функции по модулю будет изменятся больше чем на ε

Точку минимума определить из множества критических, для которой выполняется условие

а) если $AC - B^2 > 0$ и $A < 0$, то в точке M имеется максимум;

б) если $AC - B^2 > 0$ и $A > 0$, то в точке M имеется минимум;

в) если $AC - B^2 < 0$, то экстремума нет;

г) если $AC - B^2 = 0$, то вопрос о наличии экстремума остается открытым;

где $A = \left(\frac{\partial^2 f}{\partial x^2}\right)_M$, $B = \left(\frac{\partial^2 f}{\partial x \partial y}\right)_M$, $C = \left(\frac{\partial^2 f}{\partial y^2}\right)_M$, M – исследуемый эксремум.

$$f(x,y) = \frac{x^2}{10} + y^2$$

Найдём минимум.

Определим критические точки из решения системы уравнений

$$\begin{cases} \frac{\partial f}{\partial x} = 0; \\ \frac{\partial f}{\partial y} = 0; \end{cases} \quad \begin{cases} x/5 = 0; \\ 2y = 0; \end{cases} \quad \begin{cases} x = 0 \\ y = 0 \end{cases}$$

критическая т. $X_1 = (0, 0)$

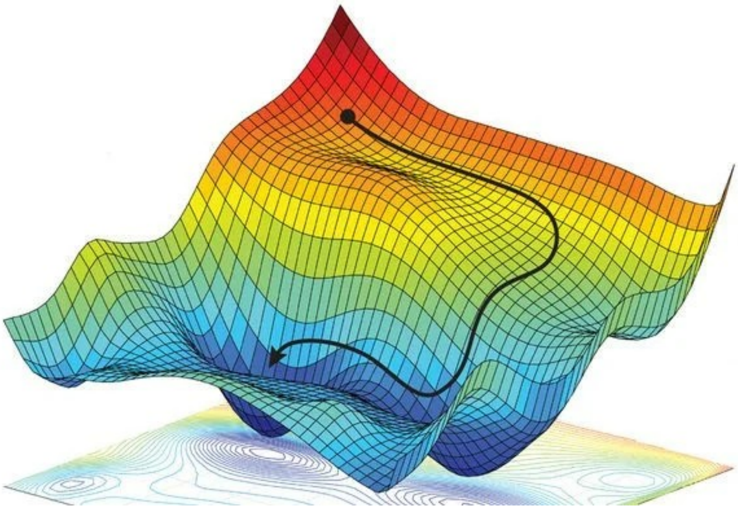
$$A = \left(\frac{\partial^2 f}{\partial x^2}\right)_{X_1} \quad B = \left(\frac{\partial^2 f}{\partial x \partial y}\right)_{X_1} \quad C = \left(\frac{\partial^2 f}{\partial y^2}\right)_{X_1}$$

$$A = 0.2 \quad B = 0 \quad C = 2$$

$$\left.\begin{array}{l} AC - B^2 = 0.4 > 0 \\ A > 0 \end{array}\right\} \Rightarrow \text{в т. } X_1 = (0, 0) \text{ – минимум функции } f$$

Ответ можно сравнить с примером численного определения минимума

Оптимизация (Mathematical optimization or mathematical programming)



Оптимизация **—** задача нахождения экстремума (минимума или максимума) целевой функции в некоторой области конечномерного векторного пространства, ограниченной набором линейных и/или нелинейных равенств и/или неравенств.

Постановка задачи

Среди элементов χ, образующих множества X, найти такой элемент χ*, который доставляет минимальное (максимальное) значение f(χ*) заданной функции f(χ).

Определим:

- Допустимое множество $\mathbb{X} = \{\vec{x} |\ g_i(\vec{x}) \leq 0,\ i = 1, \ldots, m\} \subset \mathbb{R}^n$
- Целевую функцию $f:\ \mathbb{X} \to \mathbb{R}$
- Критерий поиска (max или min).

функцию y = f (x) называют ограниченной снизу на мн-ве X , если существует такое число b , что для любого x из мн-ва X выполнено неравенство f(x) < b

Тогда решить задачу $f(x) \to \min\limits_{\vec{x} \in \mathrm{X}}$ означает одно из:

- Показать, что X = ∅
- Показать, что целевая функция $f(\vec{x})$ не ограничена снизу.
- Найти $\vec{x}^* \in \mathbb{X}:\ f(\vec{x}^*) = \min\limits_{\vec{x} \in \mathbb{X}} f(\vec{x})$
- Если $\nexists \vec{x}^*$, то найти $\inf\limits_{\vec{x} \in \mathbb{X}} f(\vec{x})$.

Инфимум (inf) подмножества X упорядоченного множества (или класса) M, называется наибольший элемент M, который равен или меньше всех элементов множества X. Привести пример инфинума для подмножества натуральных чисел

Если целевая функция не является выпуклой, то часто ограничиваются поиском локальных минимумов и максимумов: точек x_0 таких, что всюду в некоторой их окрестности выполняется $f(x) \geq f(x_0) f(x) \geq f(x_0)$ для минимума и $f(x) \leq f(x_0) f(x) \leq f(x_0)$ для максимума.

Некоторая классификация методов оптимизации

Если допустимое множество $\mathbb{X} = \mathbb{R}^n$, то такая задача называется задачей ***безусловной оптимизации***, в противном случае — задачей ***условной оптимизации***.

Задачу условной оптимизации линейной функции решает например симплекс-метод.

Если оптимизация связана с расчётом оптимальных значений параметров при заданной структуре объекта, то она называется ***параметрической оптимизацией***. Задача выбора оптимальной структуры является ***структурной оптимизацией.***

Классификация по детерминированности.

- Детерминированные;
- случайные – стохастические (например стохастический градиентный спуск);
- комбинированные.

По требованиям к гладкости и наличию у целевой функции частных производных, их также можно разделить на:

- прямые методы, требующие только вычислений целевой функции в точках приближений (например метод деформируемого многогранника);
- методы первого порядка: требуют вычисления первых частных производных функции (например градиентный спуск)
- методы второго порядка: требуют вычисления вторых частных производных, то есть гессиана целевой функции (например Бройдена — Флетчера — Гольдфарба — Шанно (BFGS))

Сходимость (convergence) – это стремление значений решения метода к соответствующим значениям решения исходной задачи при стремлении к нулю параметра дискретизации (например, шага интегрирования).

С ростом порядка метода скорость сходимости возрастает, но и возрастает вычислительная сложность каждой итерации.

Особый интерес могут для оптимизации могут представлять квадратичный функции вида

$$f(x) = f(x_1, x_2, \ldots, x_n) = \frac{1}{2} \sum_{i,j=1}^{n} a_{ij} x_i x_j - \sum_{i=1}^{n} b_i x_i + c$$

Оптимизация таких функций – одна их самых простых задач оптимизации. Если используются методы первого порядка, то для квадратичной функции получается линейный градиент

Вторая причина – эти функции часто встречаются на практике. Например при построении линейной регрессии.

Градиентный спуск (Gradient descent)

Градиентный спуск — метод первого порядка для нахождения локального экстремума (минимума или максимума) функции с помощью движения вдоль градиента.

Пусть целевая функция имеет вид:

$F(\vec{x}):\ \mathbb{X} \to \mathbb{R}$

И задача оптимизации задана следующим образом:

$F(\vec{x}) \to \min\limits_{\vec{x} \in \mathbb{X}}$

В случае, когда требуется найти максимум, вместо $F(\vec{x})$ используется $-F(\vec{x})$

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

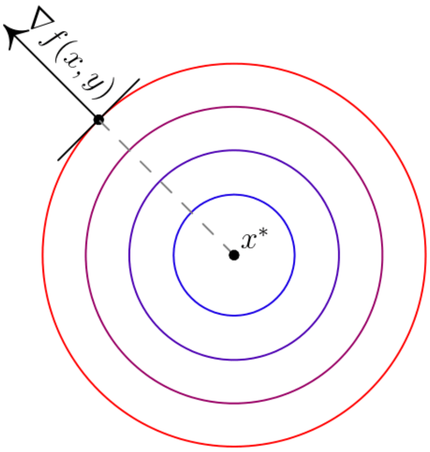
см. картинку ниже

$\vec{x}^{[j+1]} = \vec{x}^{[j]} - \lambda^{[j]} \nabla F\left(\vec{x}^{[j]}\right)$

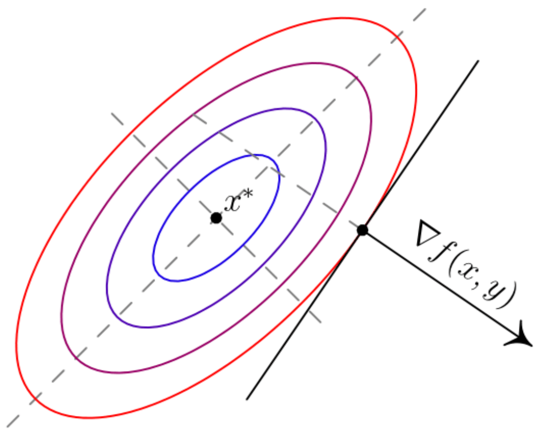
где $\lambda^{[j]}$ задает скорость градиентного спуска. Верхний индекс – номер шага.

λ может быть выбрана:

- постоянной;
- убывающей в процессе градиентного спуска;
- гарантирующей наискорейший спуск:
    1. Для поиска минимума $F(\vec{x})$ получаем $\lambda^{[j]} = \mathrm{argmin}_\lambda F\left(\vec{x}^{[j+1]}\right) = \mathrm{argmin}_\lambda\, F\left(\vec{x}^{[j]} - \lambda \nabla F\left(\vec{x}^{[j]}\right)\right)$
    2. Для поиска максимума $F(\vec{x})$ получаем $\lambda^{[j]} = \mathrm{argmax}_\lambda F\left(\vec{x}^{[j+1]}\right) = \mathrm{argmax}_\lambda\, F\left(\vec{x}^{[j]} + \lambda \nabla F\left(\vec{x}^{[j]}\right)\right)$



$f(x, y) = x^2 + y^2$



$f(x, y) = \frac{3}{2}(x - y)^2 + \frac{1}{3}(x + y)^2$

[!!!] Вектор градиента указывает на направление наискорейшего возрастания функции. Но вектор антиградиента в общем случае не указывает точно на точку минимума функции

Алгоритм

1. Задают начальное приближение и точность расчёта $\vec{x}^0$, ε
2. Рассчитывают $\vec{x}^{[j+1]} = \vec{x}^{[j]} - \lambda^{[j]} \nabla F\left(\vec{x}^{[j]}\right)$,

   где $\lambda^{[j]} = \operatorname{argmin}_\lambda F\left(\vec{x}^{[j]} - \lambda \nabla F\left(\vec{x}^{[j]}\right)\right)$

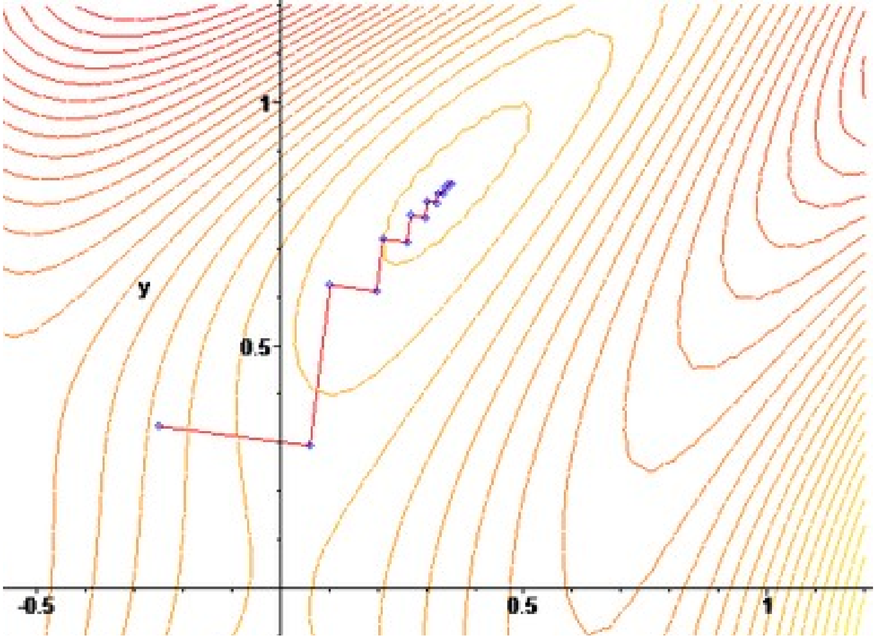
   или $\lambda^{[j]} = \lambda = const$
3. Проверяют условие остановки:
   1. Если $\left|\vec{x}^{[j+1]} - \vec{x}^{[j]}\right| > \varepsilon$, $\left|F\left(\vec{x}^{[j+1]}\right) - F\left(\vec{x}^{[j]}\right)\right| > \varepsilon$ или $\left\|\nabla F\left(\vec{x}^{[j+1]}\right)\right\| > \varepsilon$(выбирают одно из условий), то j = j + 1 и переход к шагу 2.
   2. Иначе $\vec{x} = \vec{x}^{[j+1]}$ и останов.

argmin по λ приведено для общности.

Если выбирается оптимальная длинна шага, то такой метода называется методом наискорейшего спуска.

Оптимальная длинна шага в этом случае выбирается методом одномерной оптимизации

знаком || || обозначается норма



Пример нескольких шагов метода. Причём здесь шаги метода не направлены точно в точку минимума, а приближаются к ней зигзагами.

Метафора метода: поиск спуска с горы в сильный туман, когда путь к подножью не видно.

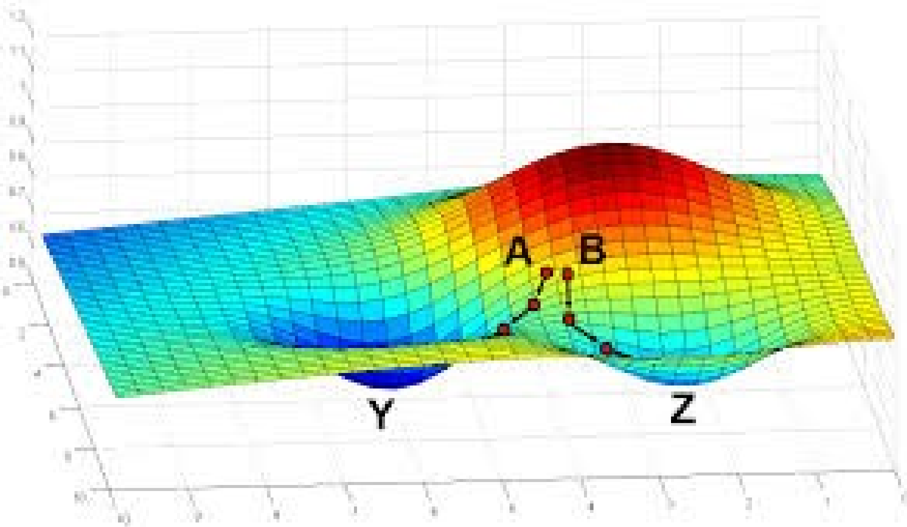
Теорема о сходимости.

k – номер шага
L – некоторое число

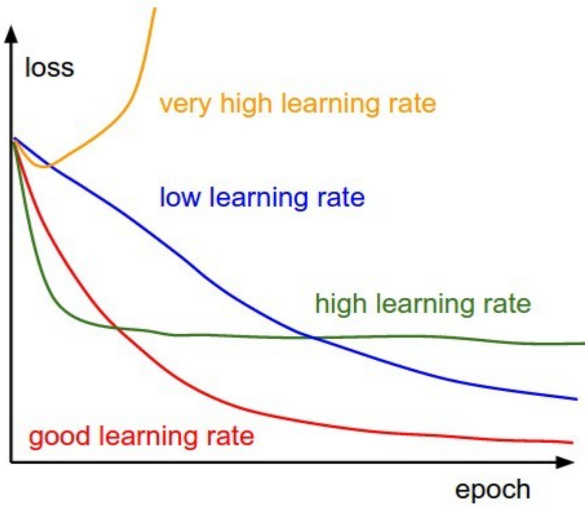
Если $\lambda = const$, $0 < \lambda < 2/L$, $f(x)$ – гладкая функция и ограничена снизу,ее градиент $f(x)$ удовлетворяет условию Липшица $\|\nabla f(x) - \nabla f(x)\| = L\|x - y\|$ для любых x и y, тогда $\lim_{k\to\infty} f'(x^{[k]}) = 0$, $f(x^{[k+1]}) < f(x^{[k]})$ при любом выборе начального приближения x_0.

Проблемы метода

- Функция должна иметь непрерывную первую производную
- Проблема медленной сходимости – проблема оврага (рисунок выше) – метод мееедленно спускается к минимуму прыгая по станкам оврага.
- Может зайти в локальный минимум (общая проблема для многих методов оптимизации)
- Нужно подбирать параметр λ : при неудачном выборе метод может сходится либо слишком медленно, либо пропускать минимум.



Найденный минимум может сильно зависеть от начальной точки



Выбор неправильного значения для длинны шага (learning rate) может сильно сказаться на минимизации функции (на примере функции ошибки – loss)

Пример. Две итерации метода градиентного спуска с постоянным шагом.

$f(x,y) = \frac{x^2}{10} + y^2$; $X_0 = (1, 1)$ $\lambda = 0.1$ $\varepsilon = 0.3$

$\text{grad } f = \left(\frac{x}{5}, 2y\right)$

итерация 1.

$X_1 = X_0 - \text{grad } f_{X_0} \cdot \lambda = (1, 1) - (0.2, 2) \cdot 0.1$

$X_1 = (0.98, 0.8)$

$|X_1 - X_0| \approx 0.2$

$|f(X_1) - f(X_0)| \approx 0.37 > \varepsilon$. продолжаем.

итерация 2

$X_2 = X_1 - \text{grad } f_{X_1} \cdot \lambda = (0.98, 0.8) - (0.196, 1.6) \cdot 0.1$

$X_2 = (0.96, 0.64)$

$|X_2 - X_1| \approx 0.026$

$|f(X_2) - f(X_1)| \approx 0.23 < \varepsilon$. останов.

Найденный минимум

$X = (0.96, 0.64)$

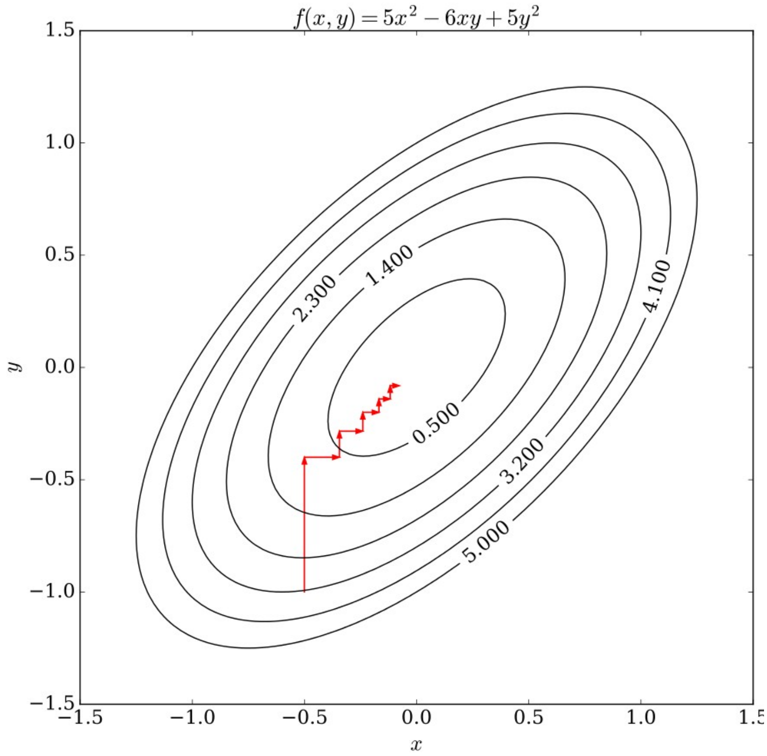
см. также
ru.wikipedia.org/wiki/Тестовые_функции_для_оптимизации

Метод покоординатного спуска (Coordinate descent)

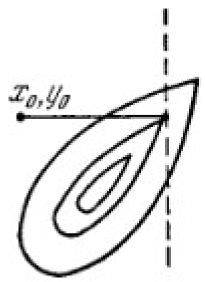
Алгоритм

1. Задают начальное приближение и точность расчёта $\vec{x}^0, \varepsilon$
2. Для всех компонент вектора $i=1 \ldots n$:
   1. Зафиксировать все компоненты вектора x кроме i-ой
   2. Решить задачу одномерной оптимизации[1] по этой компоненте, получить новое значение i-й координаты
   3. Проверить условие остановки:
   4. Если $\left|\vec{x}^{[j+1]}-\vec{x}^{[j]}\right|>\varepsilon$, $\left|F\left(\vec{x}^{[j+1]}\right)-F\left(\vec{x}^{[j]}\right)\right|>\varepsilon$ или $\left\|\nabla F\left(\vec{x}^{[j+1]}\right)\right\|>\varepsilon$(выбирают одно из условий), то j = j + 1 и переход к шагу 2 основного алгоритма.

Иначе $\vec{x}=\vec{x}^{[j+1]}$ и останов.



Шаги производятся только вдоль координат



Существуют функции, когда метод координатного спуска не приводит даже в локальный оптимум – рисунок справа.

Эффективность метода (количество итераций) зависит от

1 Можно решать задачу например методом золотого сечения или методом Ньютона для одной переменной

ориентации линий уровня относительно соей координат. В идеальном случае овраг ориентирован вдоль одной из осей координат.

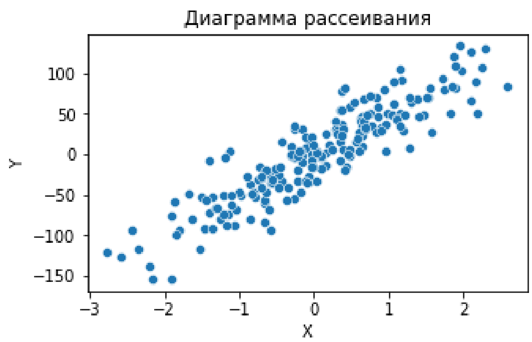
Стохастический градиентный спуск (Stochastic gradient descent, SGD)

Стохастический градиентный спуск (Stochastic gradient descent, SGD) — это метод итерации для оптимизации целевой функции с подходящими свойствами гладкости (например, дифференцируемость или субдифференцируемость). Его можно расценивать как стохастическую аппроксимацию оптимизации методом градиентного спуска, поскольку он заменяет реальный градиент (вычисленный из полного набора данных) путём оценки такового (вычисленного из случайно выбранного подмножества данных). Особенно в приложениях обработки больших данных это сокращает вычислительные ресурсы, достигая более быстрые итерации в обмен на более низкую скорость сходимости.

Математическая статистика и машинное обучение рассматривают задачу минимизации целевой функции, представленной суммой функций Q_i

$$Q(x) = \frac{1}{n}\sum_{i=1}^{n} Q_i(x)$$

Каждый член суммы Q_i обычно ассоциируется i-ым наблюдением в наборе данных (использованном для обучения).



Например Qi — ошибка на i-й точке данных в задаче построения линейной регресии

В случаях, когда вычисление градиента вышеприведённой суммы может потребовать ресурсоёмких вычислений градиентов для всех суммируемых функций стохастический градиентный спуск отбирает (случайно, поэтому метод и стохастический) подмножество суммируемых функций на каждом шаге алгоритма. Например на каждом шаге для суммы из n=100 слагаемых будет рассмотрено только 20. Такое множество называется ***батчём (batch)***.

Таким образом, алгоритм метода практически не изменяется, за исключением того, что оптимизируемая сумма на каждой итерации метода будет не полной.

по сравнению с алгоритмом градиентного спуска, здесь добавился новый пункт
↓

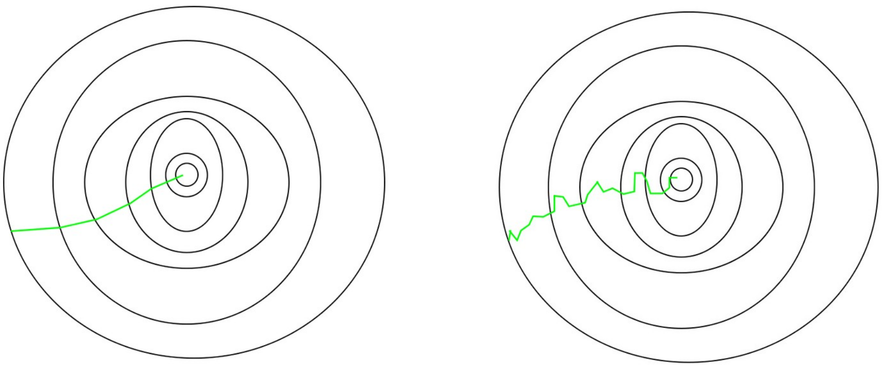
Алгоритм метода изменится следующим образом

1. Задают начальное приближение и точность расчёта $\vec{x}^0$, ε
2. Цикл по выборкам частичных сумм из n. Далее для каждой частичной суммы:
   1. Рассчитывают $\vec{x}^{[j+1]} = \vec{x}^{[j]} - \lambda^{[j]} \nabla F\left(\vec{x}^{[j]}\right)$,
3. Проверяют условие остановки:
   1. Если $\left|\vec{x}^{[j+1]} - \vec{x}^{[j]}\right| > \varepsilon$, $\left|F\left(\vec{x}^{[j+1]}\right) - F\left(\vec{x}^{[j]}\right)\right| > \varepsilon$ или $\left\|\nabla F\left(\vec{x}^{[j+1]}\right)\right\| > \varepsilon$(выбирают одно из условий), то j = j + 1 и переход к шагу 2.
   2. Иначе $\vec{x} = \vec{x}^{[j+1]}$ и останов.

Стоит отметить, что эта модификация метода призвана решить исключительно техническую проблемы – высокую ресурсоёмкость вычисления целевой функции как суммы других функций.

Под ресурсоёмкостью можно понимать высокие требования к оперативной памяти (все данных не помещаются в ОЗУ) или вычисление ошибки для всех данных занимает слишком много времени.

При этом приходится идти на компромисс – вычисленный градиент неполной суммы может плохо совпадать с градиентом целевой функции в случае полной суммы.



Пример траекторий градиентного спуска (слева) и стохастического градиентного спуска (справа)

Инерционные или ускоренные градиентные методы (momentum)

Далее в этом методе и последующих будем рассматривать алгоритм градиентного спуска, указывая только изменения в его отдельных частях.

В этих методах предлагается учитывать направление одного или нескольких предыдущих шагов:

$$p_{t+1} = \beta p_t + \nabla f(x_t)$$

$$x_{t+1} = x_t - \lambda p_{t+1}$$

← см. пункт 2 метода градиентного спуска. Здесь считаем λ шага постоянным

нижний индекс обозначает номер шага метода, *x* и *p* – векторы.

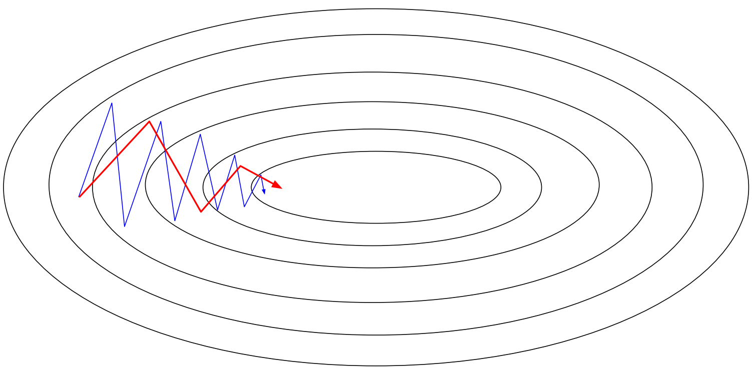
Информация о предыдущих шагах метода содержится в *p,* коэффициент β (выбирается несколько меньше единицы) отвечает за вес каждого предыдущего сделанного шага в выборе направления для текущего. Такой подход, когда каждое следующее значение рекуррентно учитывает несколько предыдущих называет *экспоненциальным сглаживанием*.

В задачах обучения нейросетей, где метод минимизирует функцию ошибки, рекомендуется выбирать значение β в диапазоне от 0.8 до 0.999.

Таким образом удаётся сгладить направление каждого из шагов, что решает проблему частой смены направления шага и как следствие решается проблема общей низкой скорости сходимости.

В дополнение к этому, удаётся перешагивать некоторые локальные минимумы, за счёт инерции шага.

Один из первых таких методов появился в середине 20 века и назывался **метод тяжелого шарика**, что передавало природу инерционности метода: в этом методе не зависят от и аккуратно подбираются в зависимости от целевой функции.



Красная ломанная — инерционный метод градиентного спуска, синяя ломанная – классический метод градиентного спуска

Показать визуализацию пути:
distill.pub/2017/momentum – демонстрация (и описание) работы метода в зависимости от параметров: длинны шага и коэффициента β

Пример

Инерционный градиентный спуск

$$f(x,y) = \frac{x^2}{10} + y^2 \qquad X_0 = (1,\ 1)$$

$$\operatorname{grad} f = \left(\frac{x}{5},\ 2y\right)$$

$$\lambda = 0.1 \qquad \beta = 0.8 \qquad \varepsilon = 0.3$$

$$P_1 = 0 + \operatorname{grad} f_{x_0}$$

$$X_1 = X_0 - \boxed{\operatorname{grad} f_{x_0}} \cdot \lambda = (1,1) - (0.2,\ 2)\cdot 0.1$$

$$X_1 = (0.98,\ 0.8)$$

$$|X_1 - X_0| \approx 0.2$$

$$|f(x_1) - f(x_0)| \approx 0.37$$

$$P_2 = \beta \cdot P_1 + \operatorname{grad} f_{x_1}$$

$$P_2 = 0.8 \cdot (0.2,\ 2) + (0.196,\ 1.6) = (0.36,\ 3.2)$$

$$X_2 = X_1 - \lambda \cdot P_2$$

$$X_2 = (0.98,\ 0.8) - 0.1 \cdot (0.36,\ 3.2) = (0.94,\ 0.48)$$

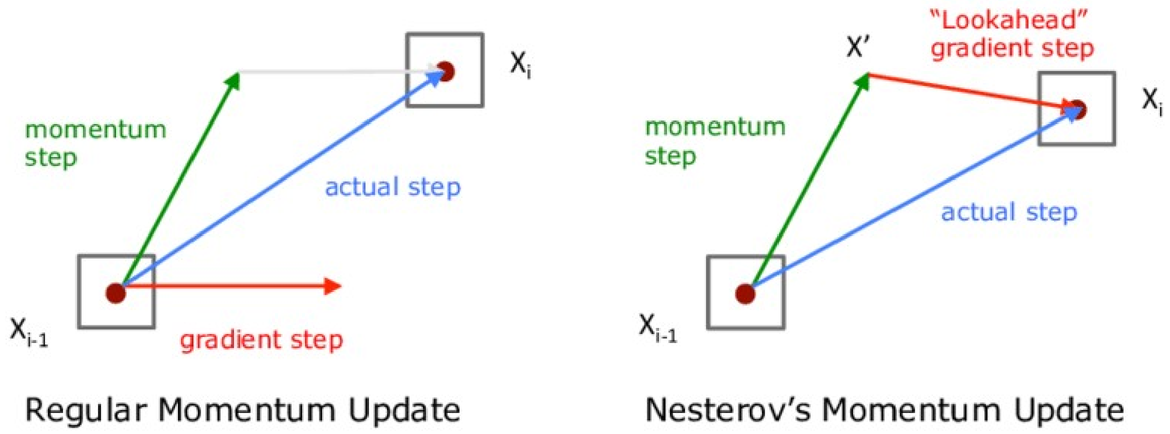
$$|X_2 - X_1| \approx 0.32$$

$$|f(x_2) - f(x_1)| \approx 0.42$$

Метод Нестерова (Nesterov Momentum)

Сравнить с аналогичным примером для градиентного спуска

Если представить, что происходит оптимизация функции двух переменных, то работу предыдущего метода можно представить левой частью рисунка ниже. Шаг, изменяющий параметры функции (синий вектор на рис. ниже), складывался из двух векторов *p* – вектор накопленных градиентов (на рисунке ниже это зелёный вектор) и градиента функции с некоторым множителем (на рисунке ниже это



красный вектор).

Причём градиент вычисляется для той же точки (x_{i-1}), откуда предполагалось сделать шаг.

В методе Нестерова предложено вычислять градиент функции не в точке x_{i-1}, а начала изменить координаты по направлению накопленного градиента (правая часть рисунка выше), и вычислить градиент уже в новой точке x'.

Таким образом обновление координат в методе градиентного спуска будет выглядеть так

$$p_{t+1} = \beta p_t + \nabla f(x_t + \beta p_t)$$

$$x_{t+1} = x_t - \lambda p_{t+1}$$

← тут, по сравнению с предыдущем случаем, поменялась только точка, в которой вычисляется градиент

Такая «корректировка» вектора p сокращает количество итераций, для отыскания минимума по сравнению с инерционным методом.

Метод сопряжённых градиентов (Nonlinear conjugate gradient method)

Некоторые определения

Пусть заданы $\vec{S_1}, \ldots, \vec{S_n} \in \mathbb{X} \subset \mathbb{R}^n$.

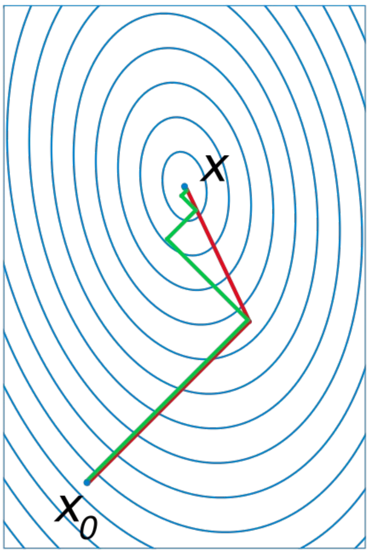
Введём на X целевую функцию $f(\vec{x}) \in \mathrm{C}^2(\mathbb{X})$

Векторы $\vec{S}_1, \ldots, \vec{S}_n$ называются сопряжёнными, если:

- $\vec{S}_i^{\,T} H \vec{S}_j = 0, \quad i \neq j, \quad i, j = 1, \ldots, n$
- $\vec{S}_i^{\,T} H \vec{S}_i \geqslant 0, \quad i = 1, \ldots, n$

где H — матрица Гессе $f(\vec{x})$ – матрица состоящая из вторых производных функции:

$$H(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1\,\partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1\,\partial x_n} \\ \frac{\partial^2 f}{\partial x_2\,\partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2\,\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n\,\partial x_1} & \frac{\partial^2 f}{\partial x_n\,\partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$



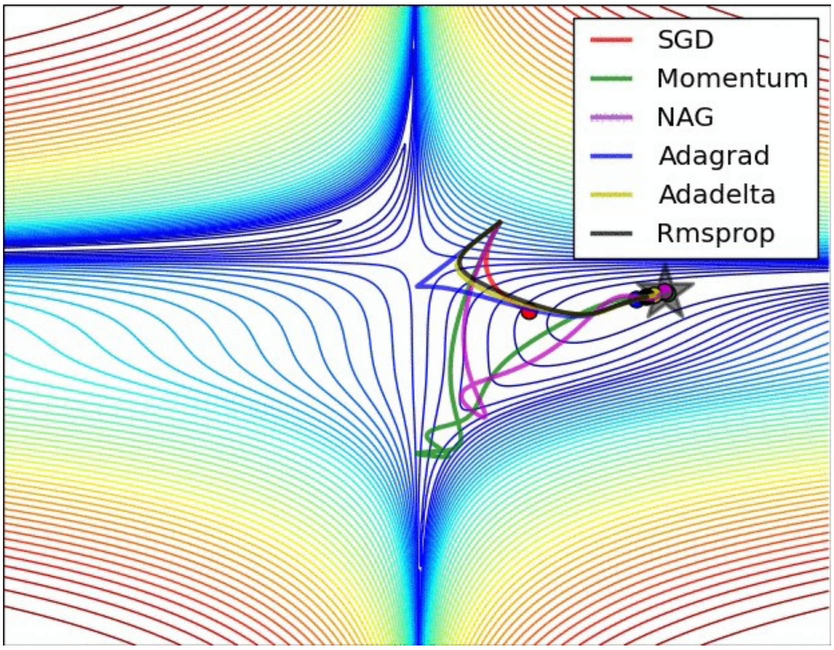
метода наискорейшего спуска – зелёная ломаная; метода сопряжённых градиентов – красная ломаная

- метод сходится быстрее чем обычный метод градиентного спуска
- при этом (как и в вышеприведённых модификациях) не требуется вычислять вторую производную.
- В случае квадратичной функции в R^n минимум находится не более чем за n шагов.

***

См. также другие модификации метода градиентного спуска

- Adam
- Adagrad
- AdaDelta, RMSProp



Путь поиска минимума для некоторых модификаций метода градиентного спуска

Ссылки

1. habr.com/ru/post/413853 – Обзор градиентных методов в задачах математической оптимизации
2. youtu.be/WZQYMGeG4xE – Обучение нейронных сетей.
3. programforyou.ru/projects/gradient-descent-optimization – живая анимация работы метода градиентного спуска и его модификаций
4. Современные численные методы оптимизации. Метод универсального градиентного спуска: Учебное пособие. — 2-е изд., испр. — М.: МЦНМО, 2021. — 272 с.
5. Пример на Python: colab.research.google.com/drive/1VuiBRDRLVFzE5SxST4KwX3Uwn3MgdIAg?usp=sharing

Задачи

Задача 1

Найдите производные функций

$$y=\frac{3^x+5}{\cos x} \quad , \quad y=\frac{1}{2}\sqrt[3]{x^2}\cos x$$

Задача 2

Найти производную функции, заданной неявно

$$3x^4y^5=e^{7x-4y}=4x^5+2y^4$$

Задача 3

Найти значение всех частных производных первого порядка в точке M(2, 1, 0)

$$u=arctg(xy^2+z)$$

Задача 4

Найти значение градиента функции $u=\sin(x+2y)+\sqrt{xyz}$ в точке

M_0 ($\frac{pi}{2}$, $3\frac{pi}{2},2$) по направлению вектора $\vec{l}$ = (4,3)

Задача 5

Опишите две итерации метода градиентного спуска для функции $f(x,y)=(x^2+y-11)^2+(x+y^2-7)^2$ для x_0 = (0,3), λ = 0.1. Производные должны быть вычислены аналитически.