MetOpt1

Дроздов Олег, Широков Данил, Исламова Камиля

1 апреля 2025 г.

1 Введение

В своей работе мы хотим найти экстремумы различных функций от двух переменных с помощью различных методов поиска минимумов и максимумов. Методы работают на основе градиентного спуска. Также мы сравним полученные результаты с встроенными библиотечными функциями из библиотеки scipy.optimize.

2 Hyperпараметры

- learning rate — это скорость обучения, которая определяет, насколько сильно обновляется значение переменной на каждом шаге градиентного спуска.
- max iterations — максимальное количество итераций, которое алгоритм будет выполнять во время поиска минимума функции
- lr method const — константа, используемая в некоторых методах выбора размера шага (например, в методе Армихо, экспоненциального уменьшения и других)
- lr method — метод для выбора размера шага. Подробнее дальше
- tolerance — пороговое значение, используемое для определения, когда алгоритм должен остановиться. Если изменение между итерациями становится меньше этого значения, считается, что алгоритм сошелся. В нашем случае мы выбрали значение = 1e-6

3 Класс градиентного спуска

3.1 Функция compute gradient

Эта функция вычисляет градиент заданной функции в определенной точке: происходит определение переменных, парсинг функции, вычисление частных производных, конвертая частных производных в функции, и в итоге вычисление самого градиента.

3.2 Функция solve

Эта функция выполняет итеративный процесс градиентного спуска для минимизации заданной функции. В основном цикле выполняется следующее (выполняется пока не достигнуто максимальное количество итераций или не выполнено условие градиентного спуска): вычисление градиента и его нормализация, определение размера шага (в зависимости от метода) и обновление точки по формуле:

$$point_k = point - step * grad \tag{1}$$

3.3 Метод plot descent

Этот метод визуализирует путь градиентного спуска в 3D. Он строит график, на котором отображаются все точки, через которые проходил алгоритм во время поиска минимума.

4 Используемые методы

4.1 Фиксированный шаг

Используется постоянное значение на всеъ итерациях

$$step = learning\_rate \tag{2}$$

4.2 Методы для адаптивного выбора шага

4.2.1 Метод Армиджо (armijo)

Вычисляются значения функции в текущей и новой точках (по формуле из solve). Если новое значение функции меньше, чем $currentf(point) - lr\_method\_const * current\_step$, то шаг уменьшается вдвое.

4.2.2 Экспоненциальное затухание (exp decay)

Этот метод уменьшает скорость обучения по экспоненциальному закону с каждой итерацией:

$$step = learning\_rate * e^{-const*iteration} \tag{3}$$

4.2.3 Затухание по времени (dec time)

Этот метод также уменьшает скорость обучения с увеличением номера итерации, но по линейному закону:

$$step = learning\_rate/1 + const * iteration \tag{4}$$

4.3 Одномерные поиски

4.3.1 Золотое сечение (golden ration)

Ищет минимум функции на заданном интервале [a, b] с использованием отношения золотого сечения

1. Шаг золотого сечения

$$\phi = \frac{sqrt(5) + 1}{2} \tag{5}$$

2. Сравниваем значение в двух точках c и d. Выбираем такой интервал, чтобы минимум оставался внутри.

$$c = b - \frac{b - a}{\phi} \tag{6}$$

$$d = a + \frac{b - a}{\phi} \tag{7}$$

3. Процесс повторяется до тех пор, пока длина отрезка не станет меньше заданного порога (epsilon).

4.3.2 Дихотомия (dichotomy)

Метод дихотомии также является методом одномерной оптимизации. Он использует подход "разделяй и властвуй"для нахождения оптимального шага на изначально заданном отрезке [a, b]

1. Находится средняя точка (m) и две точки:

$$c = m - delta; d = m + delta \tag{8}$$

2. Значения функции в этих точках сравниваются, и границы обновляются в зависимости от того, где находится минимум.

4.4 Библиотечный метод

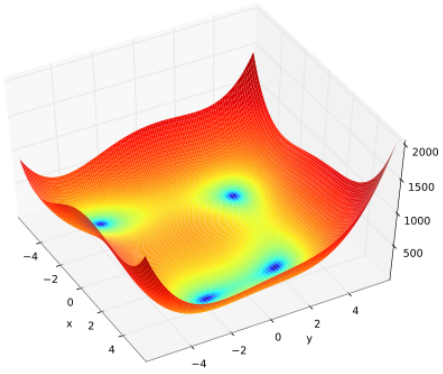
Мы решили использовать функцию minimize из библиотеки scipy.optimize (method='BFGS')

5 Исследумые функции

5.1 Функция Химмельблау

Пример мультимодальной функции. Имеет четыре равнозначных локальных минимума в точках: $(3; 2); (\approx -2,8; \approx 3,13); (\approx -3,77; \approx -3,28); (\approx 3,58; \approx -1,84)$

$$(x^2 + y - 11)^2 + (x + y^2 - 7)^2 \tag{9}$$

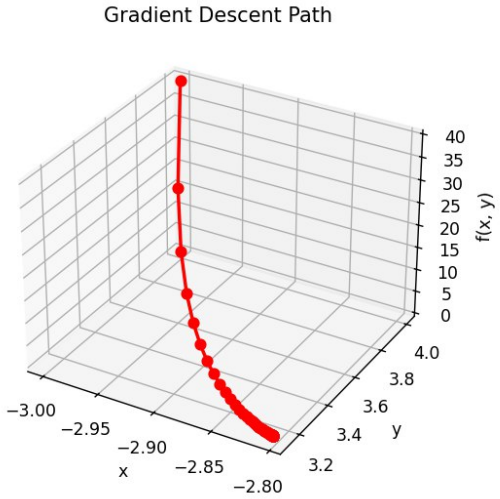


5.2 Парабалоид

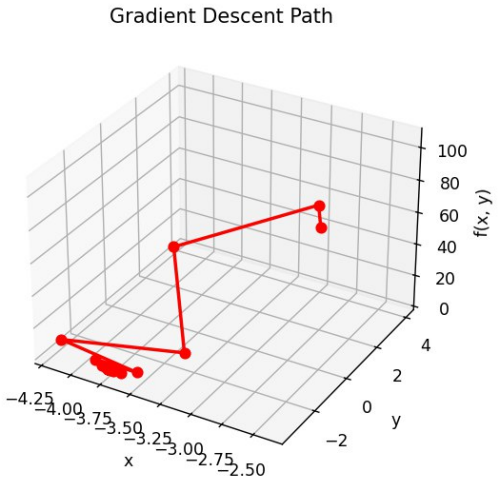
$$3(x-3)^2 + y^2 \tag{10}$$

6 Графики

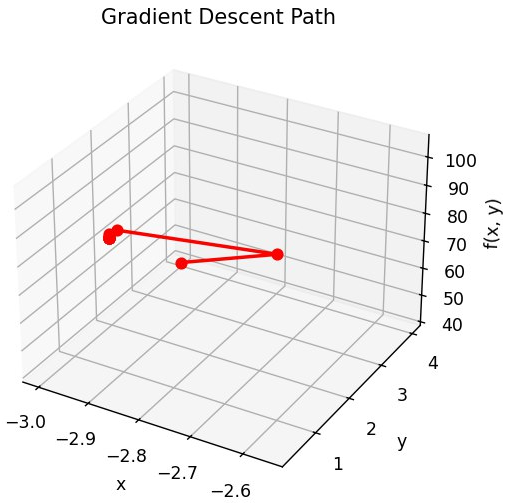
6.1 Функция Химмельблау



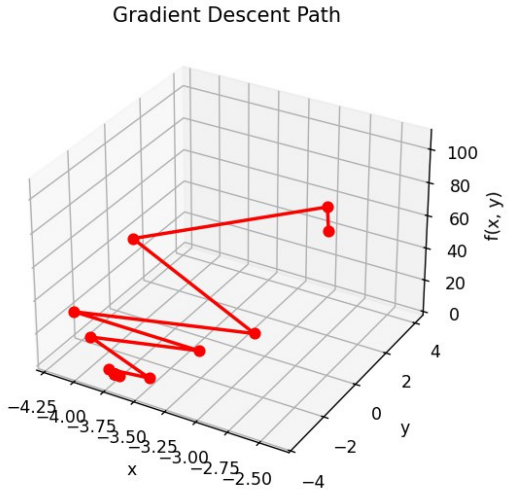
- fixed



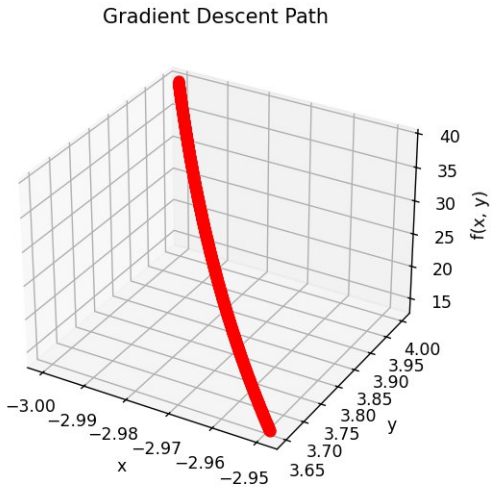
- armijo



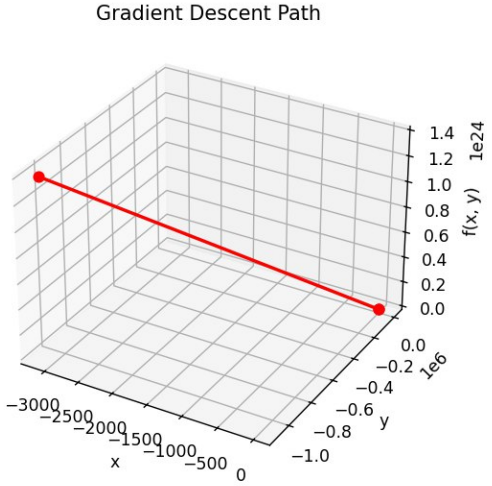
- exp decay



- dec time



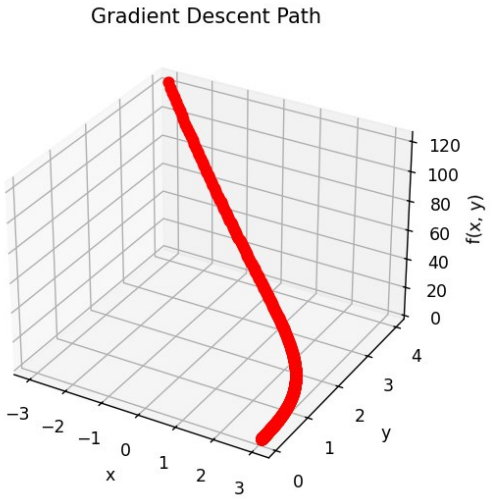
- golden ratio



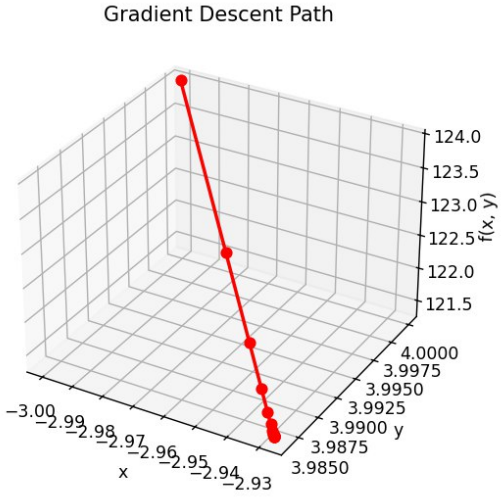
- dichotomy

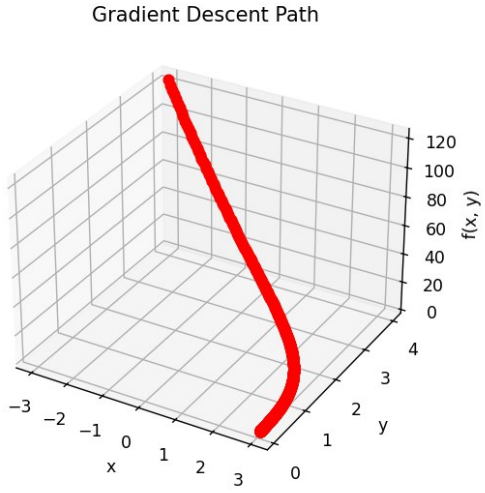
6.2 Парабалоид

- fixed

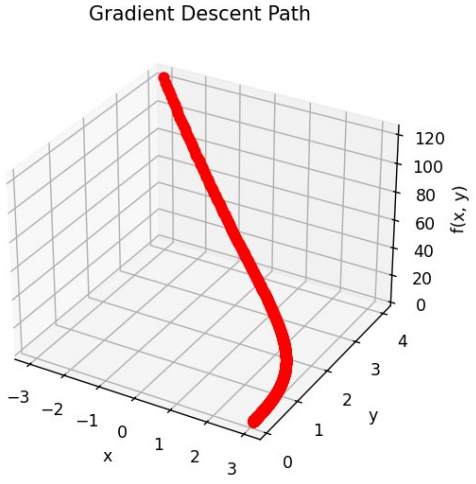


- armijo

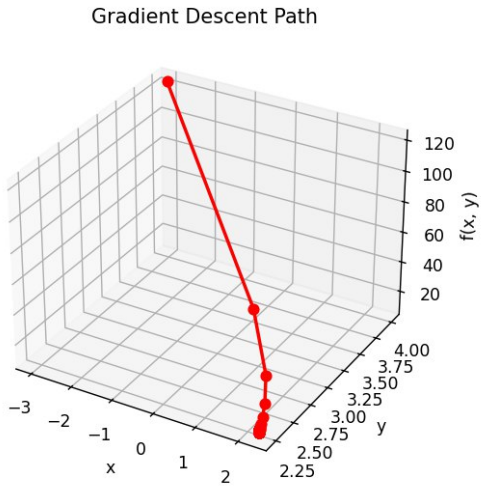




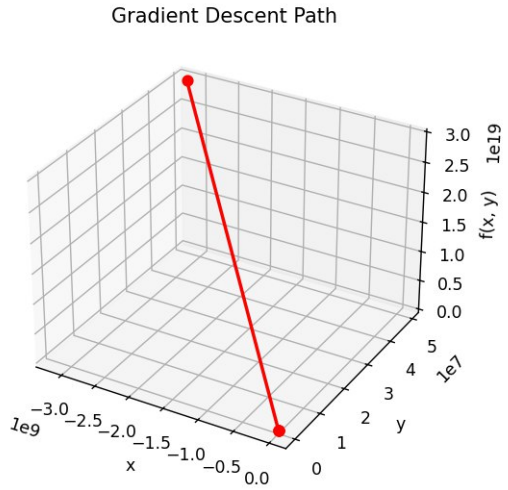
- exp decay



- dec time



- golden ratio



- dichotomy

7 Таблица

Функция	Метод	Точка	Вычисление функции	Вычисление градиента	Количество итераций
Химмельблау	Golden Ratio	(-2.9490822182637872, 3.6503812136894167)	49000	1000	1000
	Fixed	(-2.8051247595299875, 3.131314135592984)	76	76	76
	Armijo	(-3.7793106856881664, -3.283185723174792)	156	78	39
	Exp decay	(-2.9093334729299105, 0.8242661058450983)	16	16	16
	dec_time	(-3.779308675665182, -3.2831834473273935)	26	26	26
	dichotomy	(-3124.3788994471734, -1087454.0333544714)	9	3	4
	BFGS	(-2.805e+00 3.131e+00)	30	10	7
Парабалоид	Golden Ratio	(2.26633098610913, 2.243102205941972)	49000	1000	1000
	Fixed	(2.999965715656477, 0.07267723814235824)	1000	1000	1000
	Armijo	(-2.928288592182528, 3.984021564311481)	68	34	17
	Exp decay	(2.99993828179844, 0.08827599010272082)	1000	1000	1000
	dec_time	(2.999939378992544, 0.08775366819681932)	1000	1000	1000
	dichotomy	(-3152710237.1844587, 49227521.315847084)	13	5	5
	BFGS	(3.000e+00 2.721e-07)	27	9	8

8 Выводы по методам

- Фиксированный шаг. Это самый простой подход, но он может не всегда работать эффективно, особенно если функция имеет сильно наклонные участки или "плоские"области.
- Метод Армиджо позволяет адаптировать шаг в зависимости от поведения функции и помогает избежать слишком больших шагов, которые могут привести к "перепрыгиванию"минимума.
- Экспоненциальное затухание: начальная скорость обучения уменьшается с увеличением номера итерации. Это позволяет алгоритму делать большие шаги в начале, когда он еще далеко от минимума, и меньшие шаги по мере приближения к минимуму.
- Затухание по времени направлено на более точное вычисление экстремума
- Метод золотого сечения: подходит для функций, которые имеют гладкие и непрерывные кривые. Лучше использовать, когда известен диапазон значений функции.
- Метод дихотомии может быть полезен для функций с резкими изменениями или разрывами. Может быть использован в случаях, когда необходимо контролировать точность нахождения минимума.

9 Конец

Спасибо за внимание!