Отчет по лабораторной работе №1

**Методы нулевого и первого порядка**

*Аксенова Валерия, Коваленко Александр, Шустров Андрей*

***Описание методов:***

Всем методам по умолчанию установлены ограничения на максимальное количество итераций (*1000*) и точность определения сходимости (*1e-6*)

1. *Градиентный спуск со стратегией выбора шага:* использованбазовый алгоритм без оптимизаций, который принимает в виде параметра функцию вычисления размера шага. Реализованы 4 подхода:

*-Постоянный шаг*

*-Фиксированное уменьшение размера шага*

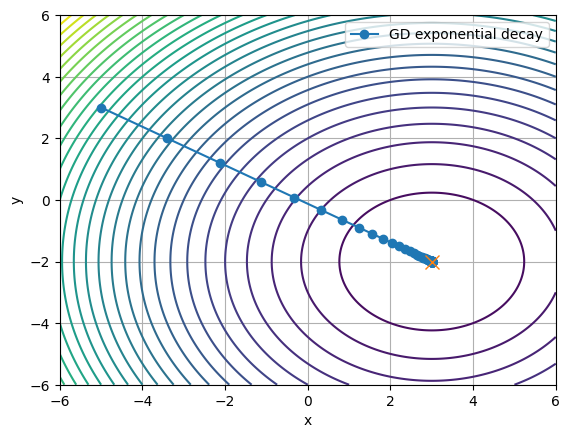
*-Экспоненциальное уменьшение размера шага*

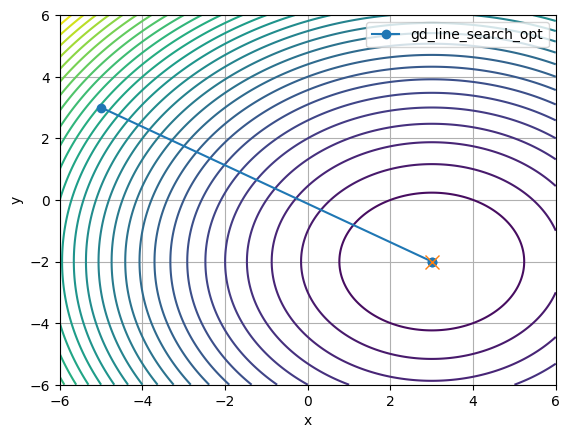
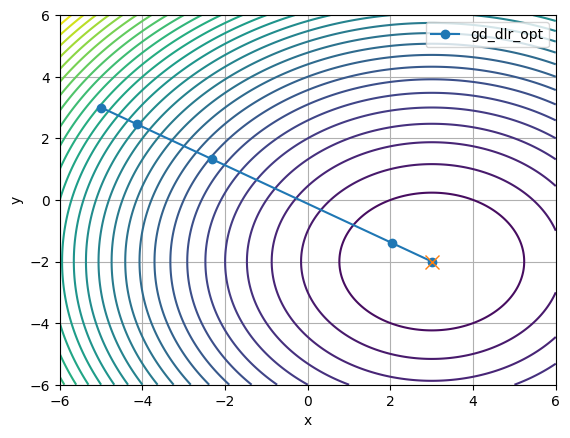
*-Изменение размера шага по косинусной кривой*

1. *Градиентный спуск, реализованный на методе золотого сечения:* использованбазовый алгоритм без оптимизаций, в котором с помощью метода золотого сечения находится оптимальный размер шага на каждой итерации.
2. *Метод BFGS из SciPy Optimize:* в библиотеке SciPy Optimize нет возможности выбора стратегии выбора шага, как в PyTorch или TensorFlow. Как аналог была использована функция ***minimize*** с методом *BFGS, так как он динамически подбирает размер шага с помощью одномерного поиска.*

1. *Градиентный спуск, реализованный на методе золотого сечения из SciPy:* использованбазовый алгоритм без оптимизаций, в котором как аналог была использована функция ***minimize\_scalar*** с методом золотого сечения (*method=‘golden’*), с помощью которого находится оптимальный размер шага на каждой итерации.
2. *Градиентный спуск, реализованный на методе интерполяции многочленом: использован* базовый алгоритм без оптимизаций, в котором с помощью метода интерполяции рядом Тейлора находится оптимальный размер шага на каждой итерации.

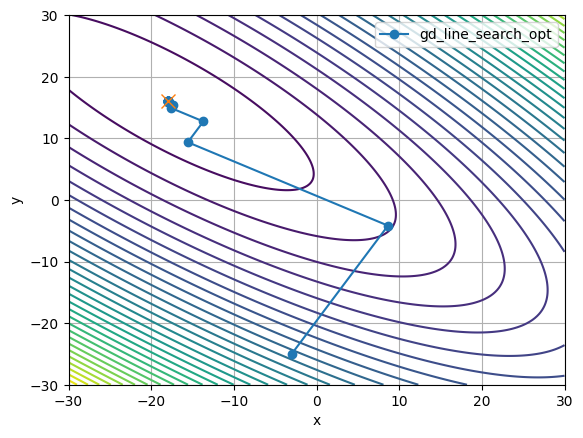
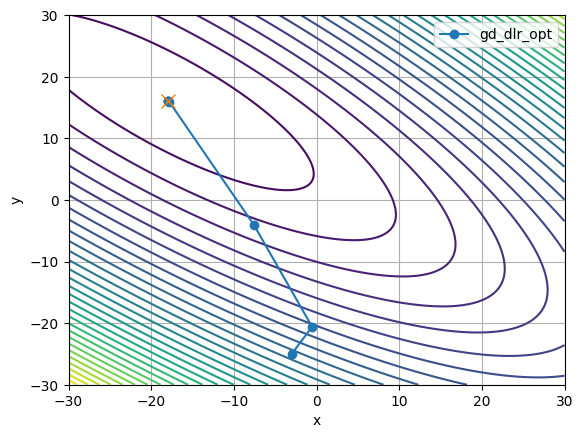
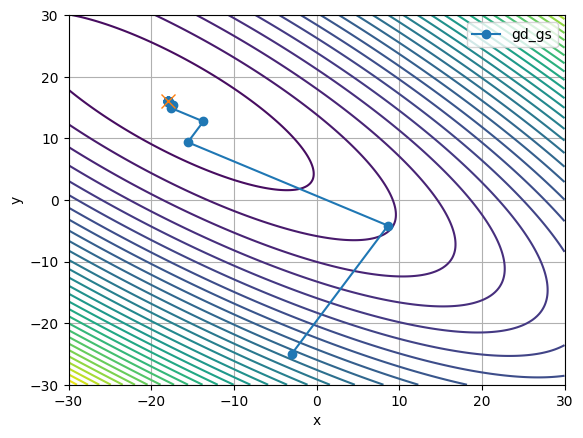
***Графики****:*

Изображение выглядит как круг, линия, диаграмма, Параллельный

Контент, сгенерированный ИИ, может содержать ошибки.Изображение выглядит как круг, линия, диаграмма, Параллельный

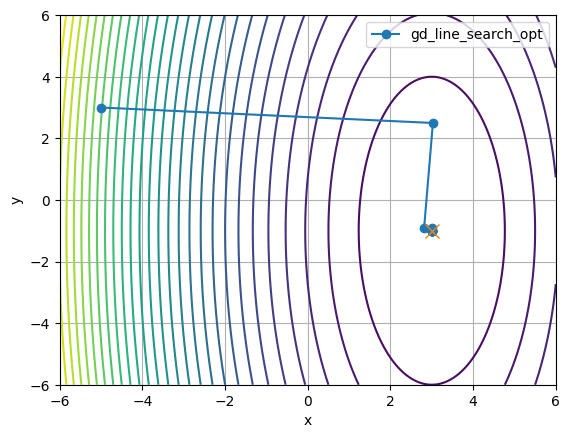
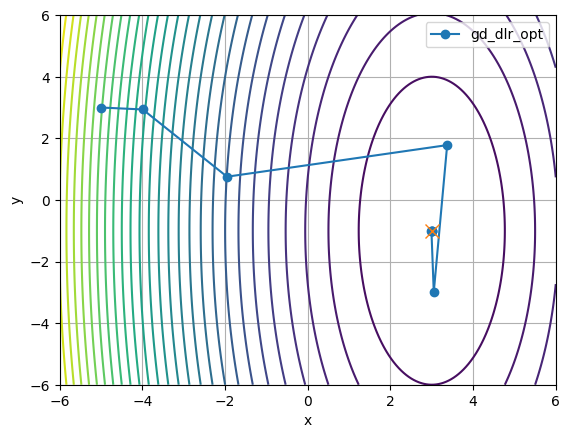
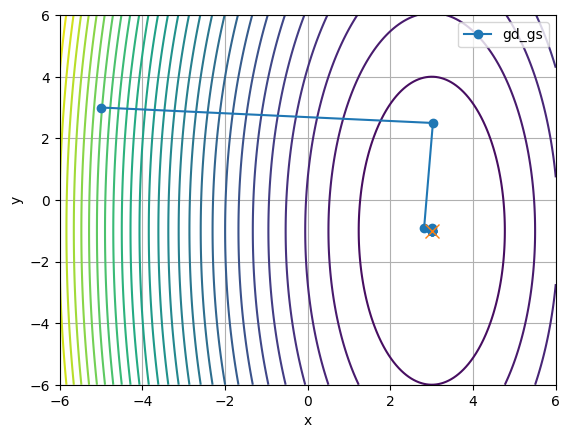
Контент, сгенерированный ИИ, может содержать ошибки.

**Изображение выглядит как линия, диаграмма, График, текст

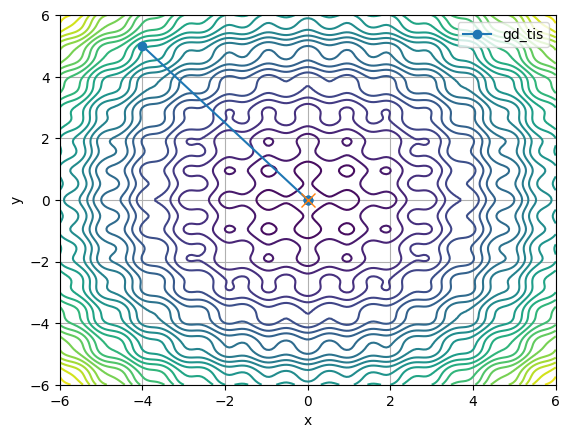
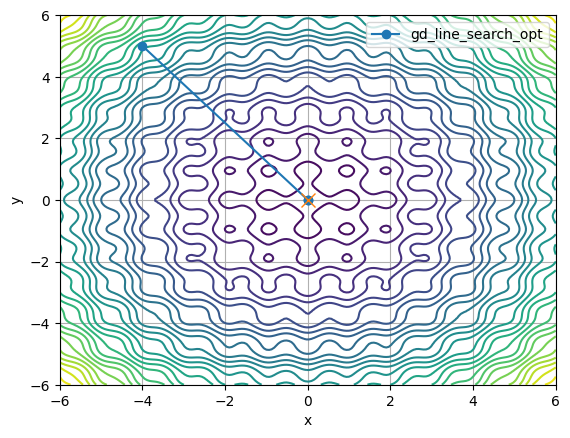
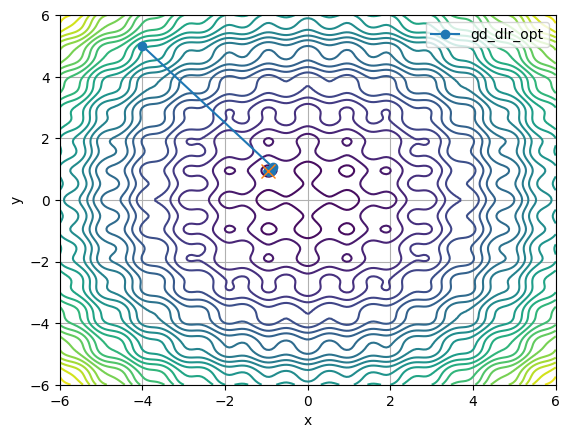
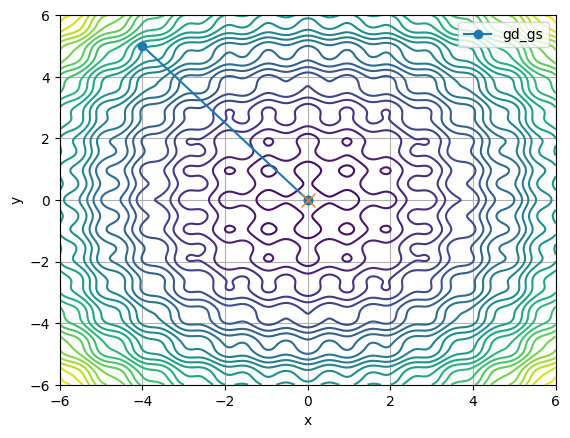
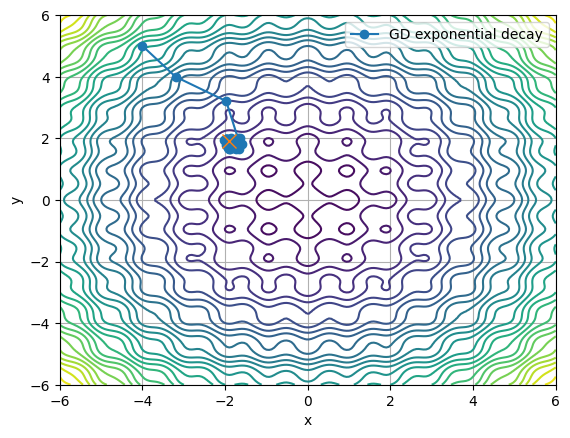
Контент, сгенерированный ИИ, может содержать ошибки.Изображение выглядит как линия, диаграмма, График, текст

Контент, сгенерированный ИИ, может содержать ошибки.**

Изображение выглядит как линия, диаграмма, текст, Параллельный

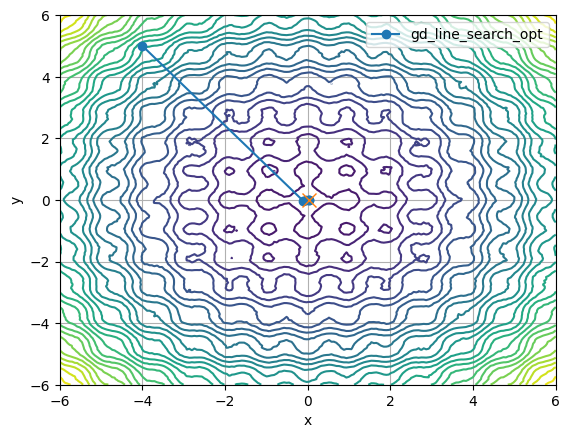
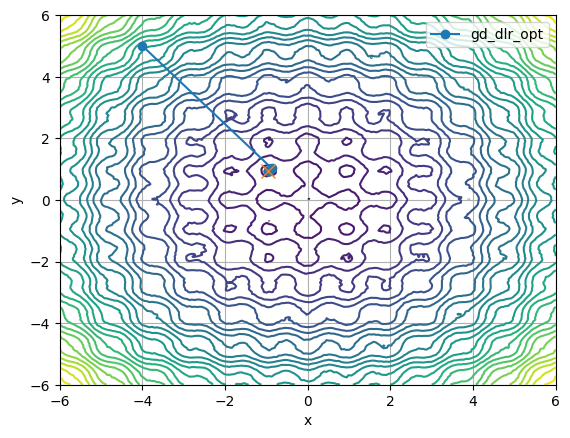
Контент, сгенерированный ИИ, может содержать ошибки.Изображение выглядит как линия, диаграмма, График, Параллельный

Контент, сгенерированный ИИ, может содержать ошибки.

****

**Изображение выглядит как шаблон, искусство, Прямоугольник, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.Изображение выглядит как шаблон, круг, снимок экрана, искусство

Контент, сгенерированный ИИ, может содержать ошибки.Изображение выглядит как снимок экрана, круг, шаблон, вода

Контент, сгенерированный ИИ, может содержать ошибки.**

***Результаты:***

Рассмотрим симметричную параболу для демонстрации базовой сходимости градиентного спуска. Градиент должен быть направлен прямо к точке минимума. Все методы запускались из начальной точки *(-5, 3).*

|  |  |  |  |
| --- | --- | --- | --- |
| *Метод* | *Кол-во итераций* | *Кол-во вычислений* | *Ошибка* |
| 1.1) const decay | 77 | 77 | 0.000000 |
| 1.2) step decay | 1000 | 1000 | 0.127724 |
| 1.3) exp decay | 153 | 153 | 0.000000 |
| 1.4) cos annealing | 166 | 166 | 0.000000 |
| 2) golden | 3 | 72 | 0.000000 |
| 3) bfgs scipy | 4 | 10 | 0.000000 |
| 4) golden scipy | 3 | 149 | 0.000000 |
| 5) taylor | 3 | 138 | 0.000000 |

Все методы показали хорошую эффективность, кроме **1.2**, в котором размер шага слишком быстро уменьшался и алгоритм не успевал дойти до минимума, что и происходило во всех последующих исследованиях. Стоит отметить, что стратегии выбора шага нуждаются в калибровке гиперпараметров для хорошей эффективности, иначе они чаще проигрывают методам, где размер шага определяется одномерным поиском, что не всегда эффективно.

Рассмотрим повернутую эллиптическую функцию. Оси симметрии не совпадают с осями координат. Все методы запускались из начальной точки *(-3, -25).*

|  |  |  |  |
| --- | --- | --- | --- |
| *Метод* | *Кол-во итераций* | *Кол-во вычислений* | *Ошибка* |
| 1.1) const decay | 190 | 190 | 0.000001 |
| 1.2) step decay | 1000 | 1000 | 6.077620 |
| 1.3) exp decay | 1000 | 1000 | 0.004507 |
| 1.4) cos annealing | 370 | 370 | 0.000001 |
| 2) golden | 20 | 701 | 0.000001 |
| 3) bfgs scipy | 6 | 16 | 0.000000 |
| 4) golden scipy | 22 | 1601 | 0.000000 |
| 5) taylor | 21 | 1516 | 0.000001 |

В результатах исследования этой функции интересно обратить внимание на результаты **2** и **4**. Наша реализация градиентного спуска на основе поиска методом золотого сечения, сейчас и далее, выполняет в 2 раза меньше вычислений функции. Вероятно, реализация из SciPy честно выполняет перерасчет для повышения точности, вместо повторного использования заранее вычисленных значений.

Также кажется важным крайне эффективный результат алгоритма **3**. Это происходит из-за того, что BFGS основывается на идее апроксимации гессиана, который является константным для квадратичных функций, что позволяет эффективно предсказывать поведение градиента.

Рассмотрим эллиптическую функцию с различной масштабностью. Функция плохо обусловлена, что может вызвать зигзагообразные скачки в пути градиентного спуска. Все методы запускались из начальной точки *(-5, 3).*

|  |  |  |  |
| --- | --- | --- | --- |
| *Метод* | *Кол-во итераций* | *Кол-во вычислений* | *Ошибка* |
| 1.1) const decay | 73 | 73 | 0.000000 |
| 1.2) step decay | 1000 | 1000 | 0.054155 |
| 1.3) exp decay | 136 | 136 | 0.000000 |
| 1.4) cos annealing | 161 | 161 | 0.000000 |
| 2) golden | 10 | 351 | 0.000000 |
| 3) bfgs scipy | 8 | 18 | 0.000000 |
| 4) golden scipy | 11 | 791 | 0.000000 |
| 5) taylor | 11 | 742 | 0.000000 |

Для данной функции можно явно заметить на графиках зигзагообразное поведение пути градиентного спуска, что обуславливается слишком большим начальным размером шага.

В качестве мультимодальной функции была выбрана функция Растригина, используемая для тестирования алгоритмов оптимизации. Данная функция обладает параметром мультимодальности **A**, который был задан 1 в нашем исследовании. Все методы запускались из начальной точки (-4, 5)

|  |  |  |  |
| --- | --- | --- | --- |
| *Метод* | *Кол-во итераций* | *Кол-во вычислений* | *Ошибка* |
| 1.1) const decay | 1000 | 1000 | 0.410774 |
| 1.2) step decay | 25 | 25 | 1.344989 |
| 1.3) exp decay | 88 | 88 | 2.682523 |
| 1.4) cos annealing | 75 | 75 | 0.000000 |
| 2) golden | 3 | 72 | 0.000000 |
| 3) bfgs scipy | 7 | 26 | 1.344989 |
| 4) golden scipy | 3 | 163 | 0.000000 |
| 5) taylor | 3 | 208 | 0.000000 |

Хорошая точность некоторых методов является результатом удачного выбора начальной точки, которая позволяет проскакивать локальные минимумы в начале пути.

Мультимодальная функция Растригина с параметром зашумленности **σ** позволяет меньше завесить от удачного выбора начальной точки благодаря внесенному шуму.

|  |  |  |  |
| --- | --- | --- | --- |
| *Метод* | *Кол-во итераций* | *Кол-во вычислений* | *Ошибка* |
| 1.1) const decay | 1000 | 1000 | 0.288458 |
| 1.2) step decay | 1000 | 1000 | 1.344989 |
| 1.3) exp decay | 1000 | 1000 | 1.896952 |
| 1.4) cos annealing | 1000 | 1000 | 0.224912 |
| 2) golden | 1000 | 35 001 | 1.033094 |
| 3) bfgs scipy | 5 | 42 | 1.351300 |
| 4) golden scipy | 5 | 389 | 0.041216 |
| 5) taylor | 1000 | 93 636 | 1.323324 |

Примечательным является результат метода **1.4**, который продолжает давать наилучшую эффективность при увеличении мультимодальности и зашумленности, а также удалении начальной точки от точки глобального минимума. Это происходит из-за изменения размера шага по косинусной кривой, что позволяет периодически “разогреваться” алгоритму и выходить из локальных минимумов.