Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт

Кафедра «Информатика»
кафедра

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 9

Метод наискорейшего градиентного спуска
Тема

Преподаватель			В. В. Тынченко
		Подпись, дата	Инициалы, Фамилия
Студент	КИ19-17/1Б, №031939174		А. К. Никитин
	Номер группы, зачетной книжки	Подпись, дата	Инициалы, Фамилия

Красноярск 2021

1 Постановка задачи

Разработать программу, реализующую метод наискорейшего градиентного спуска.

Найти безусловный экстремум функции, выбранной в соответствии с заданием, с использованием разработанной программы.

Функция: $(x_2 + x_1 - 1)^2 + 2(x_1 - 2)^2 \to min$

2 Описание метода

Градиентный спуск — метод нахождения локального минимума (максиму ма) функции с помощью движения вдоль градиента. Для минимизации функции в направлении градиента используются методы одномерной оптимизации, нап ример, метод золотого сечения. Также можно искать не наилучшую точку в нап равлении градиента, а какую-либо лучше текущей.

Сходимость метода градиентного спуска зависит от отношения максимал ьного и минимального собственных чисел матрицы Гессе в окрестности миним ума (максимума). Чем больше это отношение, тем хуже сходимость метода.

Пусть целевая функция имеет вид:

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

И задача оптимизации задана следующим образом:

$$f(\vec{x}) \to \min_{\vec{x} \in \mathbb{X}}$$

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

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

где λ[j] вычисляется по формуле:

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

3 Исходные тексты программ

На листинге 1 представлен код программы, реализующий задание.

Листинг 1 – Метод градиентного спуска с постоянным шагом

```
import numpy as np
from sympy.solvers import solve
from sympy import Symbol
from const import func

t = Symbol('t')


def calculate_step(f, x, gradient):
    t_function = f.calc(x - t * gradient)
    return solve(t_function.diff(t), t)[0]


def steepest_gradient_descent_method(f, x0, epsilon1=0.1, epsilon2=0.1, M=100):
    x_list = [np.array(x0).astype(float)]
    k = 0
    while k < M:
        gradient = f.gradient_value(x_list[k])
        if np.linalg.norm(gradient) < epsilon1:
            return x_list[k], k

        step = float(calculate_step(f, x_list[k], gradient))

        x_list.append(x_list[k] - step * gradient)

        if np.linalg.norm(x_list[k + 1] - x_list[k]) < epsilon2 \
                and abs(f.calc(x_list[k + 1]) - f.calc(x_list[k])) < epsilon2 \
                and len(x_list) > 2 \
                and np.linalg.norm(x_list[k] - x_list[k - 1]) < epsilon2 \
                and abs(f.calc(x_list[k]) - f.calc(x_list[k - 1])) < epsilon2:
            return x_list[k + 1], k

        k += 1

    return x_list[-1], k
```

Окончание листинга 1

```
if __name__ == '__main__':
    print(steepest_gradient_descent_method(func, [0.5, 1], epsilon1=0.1,
epsilon2=0.15, M=10))
```

4 Исследование влияния параметров метода на точность и скорость нахождения решения

В качестве гиперпараметров по умолчанию использовались следующие значения:

а) x0 = (-10, 10);

б) $\varepsilon_1 = 0.1$;

в) $\varepsilon_2 = 0.1$;

г) M = 100.

Из рисунка 1 можно заключить, что, в целом, чем меньше параметр ε_1, тем точнее решение, однако тем больше шагов необходимо алгоритму для нахождения ответа.

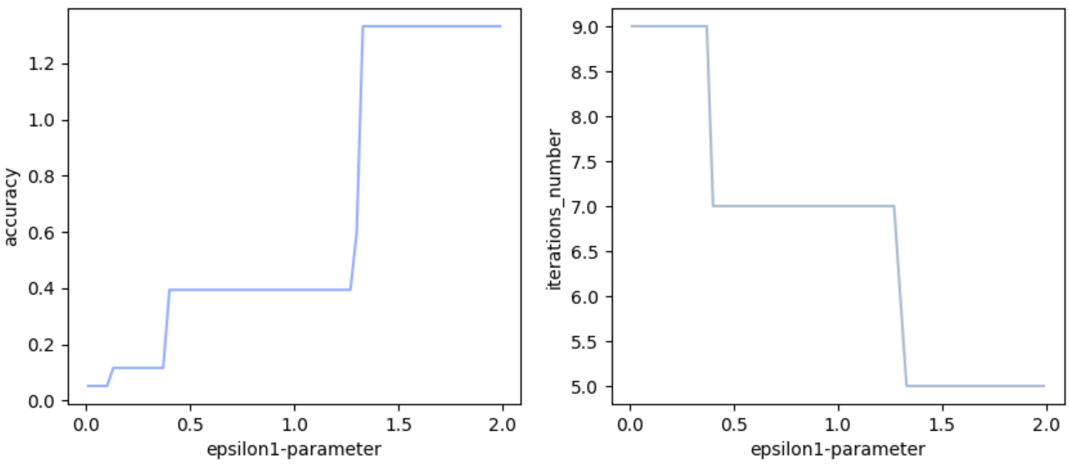

Рисунок 1 – Влияние параметра ε_1 на точность и производительность

Из рисунка 2 можно провести аналогичные рассуждения касательно параметра ε_2.

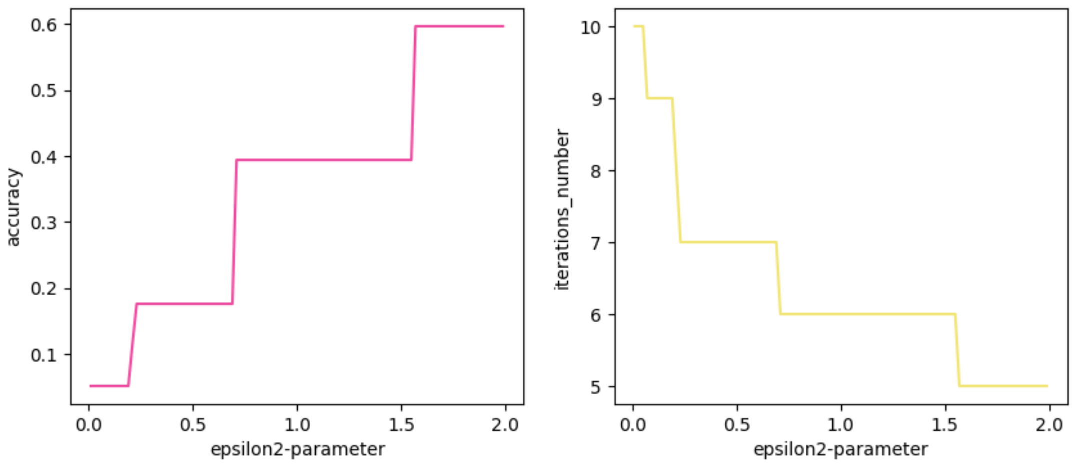

Рисунок 2 - Влияние параметра ε_2 на точность и производительность

Из рисунка 3 можно заключить, что параметр максимального количества шагов лучше не делать слишком небольшим, и что чем он больше, тем точнее, но менее производительнее, программа.

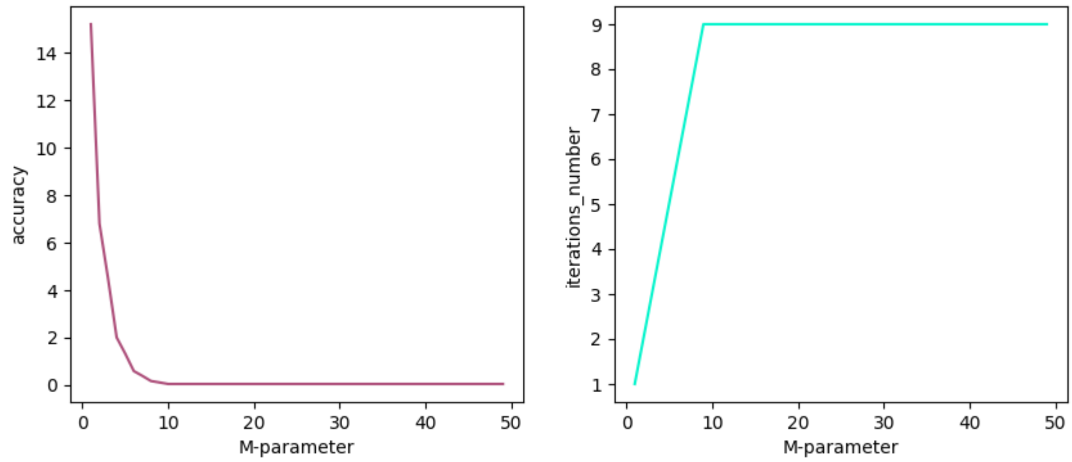

Рисунок 3 - Влияние параметра максимального количества шагов на точность и производительность

5 Вывод

В результате данной работы был реализован и проанализирован метод наискорейшего градиентного спуска для поиска локального минимума многомерной функции. Также были проанализированы гиперпараметры метода и их влияние на работу функции.