Практическое задание
«Методы оптимизации»

Необходимые файлы для выполнения практической доступны по ссылке

1. **Задание** (23 балла)

   Реализовать подсчёт следующих функций (а также их градиентов и гессианов):

   (a) $f(x) = x^2$ (class f1) (1+1+1 баллов)

   (b) $f(x) = \sin(3 \cdot \sqrt{x^3} + 2) + x^2$ (class f2) (1+1+1 баллов)

   (c) $f(x, y) = \frac{(x-3.3)^2}{4} + \frac{(y+1.7)^2}{15}$ (class f3) (1+1+1 баллов)

   (d) $f(\vec{x}) = \|\vec{x}\|^2$ (class SquaredL2Norm) (1+1+1 баллов)

   (e) $f(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2$ (class Himmelblau) (1+1+1 баллов)

   (f) $f(\vec{x}) = \sum_{i=1}^{n-1} \left[100 \cdot \left(x_{i+1} - x_i^2\right)^2 + (1 - x_i)^2\right]$ (class Rosenbrok) (2+3+3 баллов)

   В качестве ответа на это задание принимаются реализованные классы из файла submission.py

   Пример шаблонов классов:

```python
class f1:
    def __call__(self, x: float):
        """
        Args:
            x: float
        Returns:
            float
        """
        ...
        return

    def grad(self, x: float):
        """
        Args:
            x: float
        Returns:
            float
        """
        ...
```

```
        return

    def hess(self, x: float):
        """
        Args:
            x: float
        Returns:
            float
        """
        ...
        return

class Himmelblau:
    def __call__(self, x: np.ndarray):
        """
        Args:
            x: numpy array of shape (2,)
        Returns:
            float
        """
        ...
        return

    def grad(self, x: np.ndarray):
        """
        Args:
            x: numpy array of shape (2,)
        Returns:
            numpy array of shape (2,)
        """
        ...
        return

    def hess(self, x: np.ndarray):
        """
        Args:
            x: numpy array of shape (2, 2)
        Returns:
            numpy array of shape (2, 2)
        """
        ...
        return

```

Пример работы классов:

```
>>> func = f1()
>>> print(func(3.0))
9.0
>>> print(func.grad(1.2))
2.4
>>> print(func.hess(-2.4))
2.0

```

2. **Задание** (10 баллов)

Реализовать функцию minimize из submission.py - функция для реализации градиентной оптимизации. (5 баллов за градиентный спуск + 5 баллов за метод Ньютона)

Бонусное задание

В файле blackboxfuntion.py (который вам не предоставляется) находится функция black_box_function, которая принимает на вход десятимерный вектор из вещественных чисел и возращает одно действительное число.

```
def black_box_function(x: np.ndarray) -> float:
    """
    Unknown function, to be minimized
    Args:
        x: np.ndarray (x.shape = (10,))
    Returns:
        value of an unknown function at point x (float)
    """

    # does something here
    return #float value
```

Вам предоставляются файлы blackbox_run.py и blackbox_optimize.py. В blackbox_run.py реализован алгоритм обращения к black_box_function через blackbox_optimize функцию из blackbox_optimize.py. В blackbox_run.py реализовывать ничего не нужно, ваша задача - реализовать функцию blackbox_optimize - она должна по истории точек (args_history) и по истории значений функции black_box_function в этих точках (func_vals_history) выбирает, какую следующую точку следует проверить. Цель - найти минимум функции black_box_function.

Описание предоставленных файлов:

submission.py	Содержит шаблоны классов и функций для 1 и 2 задания
blackbox_run.py	Содержит процедуру обращения к функции black_box_function. Этот же файл является проверочным для бонусной задачи
blackbox_optimize.py	Содержит функцию blackbox_optimize, которую необходимо реализовать в бонусном задании
test_submission.py	Файл для проверки реализованных в submission.py функциях и классах
Dockerfile	Для сборки докер образа для запуска проверки с фиксированными пакетами
test.sh	Запускает test_submission.py и blackbox_run.py.

Проверить ваше решение можно, запустив докер:

1. Переходим в папку с вашим решением (папка должна содержать все описанные выше файлы)
2. docker build -t msu_prac_optimization .
3. docker run –rm -v $PWD:/prac_folder msu_prac_optimization

В результате в вашей папке создастся 2 файла: main_task_score.txt и blackbox_function_score.txt