

Отчет

Лабораторная работа №3

Введение

В этой лабораторной работе мы реализовали и исследовали методы стохастической оптимизации поиска минимума многомерных функций, а также сравнили их с методами из прошлых лабораторных работы. В работе были использованы Python 3 и библиотека `scipy-optimize`.

1 Описание методов

SGD

Реализация SGD

- Метод возвращает координаты и значение точки, а также историю вычислений
- для инициализации состояния применяется NumPy
- Гиперпараметры отвечают за ускорение сходимости и за остановку вычислений

Листинг 1: Реализация метода SGD

```
1 def stochastic_gradient_descent_2d(  
2     objective_func: Callable,  
3     gradient_func: Callable,  
4     x_bounds: Tuple[float, float],  
5     y_bounds: Tuple[float, float],  
6     learning_rate: float = 0.01,  
7     max_iter: int = 10000,  
8     momentum: float = 0.9,  
9     tol: float = 1e-6,  
10    random_state = None  
11 ):  
12     current_x = random.uniform(x_bounds[0], x_bounds[1])  
13     current_y = random.uniform(y_bounds[0], y_bounds[1])  
14     current_value = objective_func([current_x, current_y])  
15     best_x, best_y = current_x, current_y  
16     best_value = current_value  
17     history = [current_value]  
18  
19     velocity = np.zeros(2)  
20  
21     for k in range(max_iter):  
22         grad = gradient_func([current_x, current_y])  
23  
24         velocity = momentum * velocity - learning_rate * grad  
25  
26         current_x += velocity[0]  
27         current_y += velocity[1]  
28  
29         current_x = max(x_bounds[0], min(x_bounds[1], current_x))  
30         current_y = max(y_bounds[0], min(y_bounds[1], current_y))  
31  
32         current_value = objective_func([current_x, current_y])  
33         history.append(current_value)  
34  
35         if current_value < best_value:  
36             best_x, best_y = current_x, current_y  
37             best_value = current_value  
38  
39         if k > 10 and abs(history[-2] - history[-1]) < tol:  
40             break  
41  
42     return (best_x, best_y), best_value, history
```

2 Графики

Реализуем отображение графиков на Python, который:

- отображает визуализацию 3D
- отрисовывает траекторию градиентного спуска
- отображает линии уровня функции

Используемые библиотеки

- `numpy` — работа с массивами данных
- `matplotlib.pyplot` — создание 3D-графиков
- `matplotlib.colors.LightSource` — создание освещения для 3D-графиков

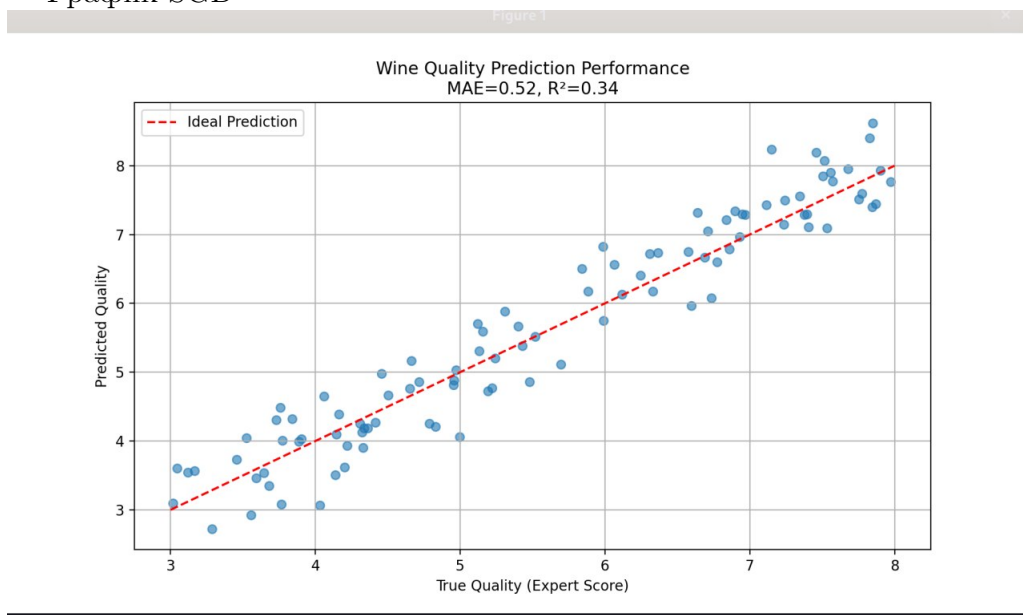
3 Описание результатов

а примере ункции иммеллау:

$$z = (x^2 + y - 11)^2 + (x + y^2 - 7)^2 \text{ в точке } (2, -1.5)$$

| Метод | Итерации | Вызовы функции | Вызовы градиента | х | у |
|-------|----------|----------------|------------------|--------------------|---------------------|
| VFSA | 1000 | 1001 | 0 | 3.5844283018777903 | -1.8481265843447385 |
| SGD | 135 | 136 | 135 | 3.5845214196109647 | -1.8482757702034391 |

График SGD

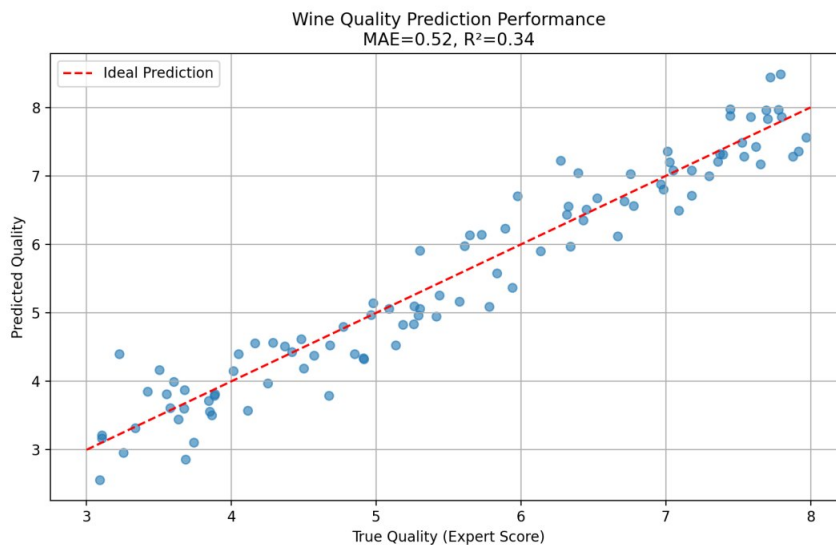


На примере функции Химмельблау:

$$z = (x^2 + y - 11)^2 + (x + y^2 - 7)^2 \text{ в точке } (-5, 0)$$

| Метод | Итерации | Вызовы функции | Вызовы градиента | х | у |
|-------|----------|----------------|------------------|---------------------|---------------------|
| VFSA | 1000 | 1001 | 0 | 3.5844284293403588 | -1.8481265430393075 |
| SGD | 156 | 157 | 156 | -2.8053155593848946 | 3.131665994249259 |

График VFSA

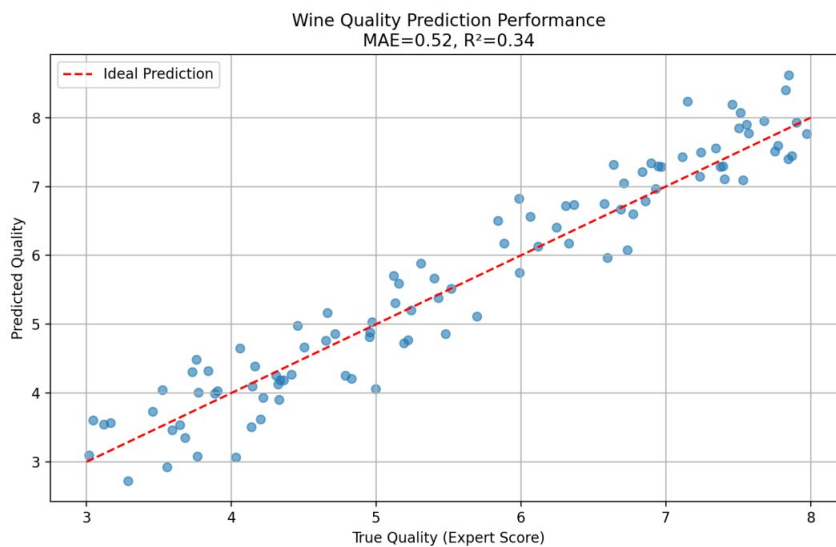


На примере функции Бута

$$z = (x + 2y - 7)^2 + (2x + y - 5)^2 \text{ в точке } (-1.5, 1.75)$$

| Метод | Итерации | Вызовы функции | Вызовы градиента | x | y |
|-------|----------|----------------|------------------|--------------------|--------------------|
| VFSA | 1000 | 1001 | 0 | 1.0004717945794974 | 2.9994346757115355 |
| SGD | 214 | 215 | 214 | 0.9970054058881611 | 3.0030057145346216 |

График SGD

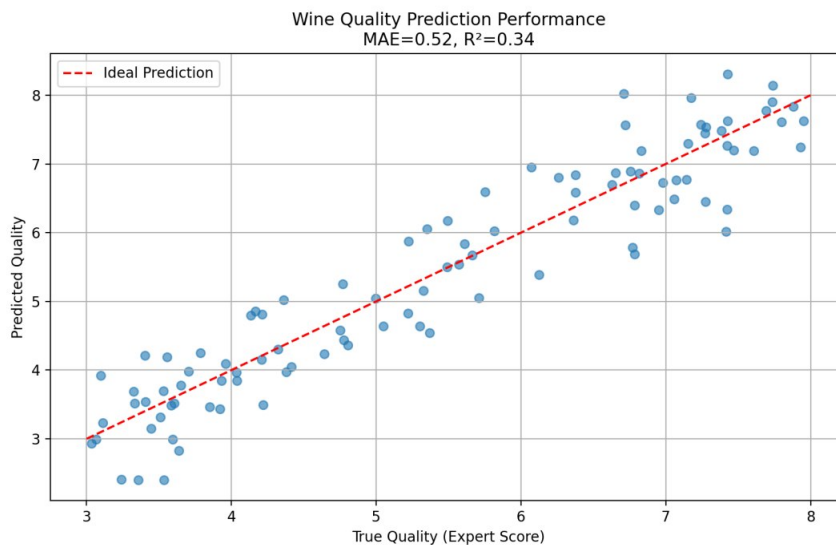


На примере функции Бута

$$z = (x + 2y - 7)^2 + (2x + y - 5)^2 \text{ в точке } (1, 0.5)$$

| Метод | Итерации | Вызовы функции | Вызовы градиента | x | y |
|-------|----------|----------------|------------------|--------------------|--------------------|
| VFSA | 1000 | 1001 | 0 | 1.0002014585959855 | 2.9998479101489495 |
| SGD | 182 | 183 | 182 | 1.0029391954868798 | 2.9972734777757633 |

График VFSA



4 Вывод

В данной лабораторной работе мы реализовали и сравнили два стохастических метода оптимизации VFSA и SGD. VFSA показал стабильные результаты даже при разных начальных точках и хорошо справляется с интересными мультимодальными функциями, при этом не требует градиента, но делает больше вызовов функции. SGD работает быстрее, но чувствителен к выбору стартовой точки и параметров, особенно скорости обучения, и может не сойтись на сложных функциях.