

Дополнительное задание 1

Мы протестировали optuna на функциях из первой лабораторной работы:

1) $x^{**2} + (x - 5)^{**2} + (2 * x - 4 * y)^{**2}$

2) $x^{**2} - x * y + 2 * x + y^{**2} - 4 * y + 3$

3) $(1 - x)^{**2} + 100 * (y - x^{**2})^{**2}$

Полученная погрешность:

1) 0.0003693441390737462

2) $2.027238616975069e-06$

3) 0.0007984539416227669

Чтобы оценить полученный результат давайте вспомним, какой погрешности нам удалось добиться используя классические методы рассмотренные в первой лабораторной работе:

	NELDER-MEAD	GRADIENT DESCENT	NEWTON	QUASI-NEWTON
1	$10^{**}(-9)$	$10^{**}(-7)$	0	0
2	$10^{**}(-1)$	$10^{**}(-1)$	0	0
3	-	-	$10^{**}(-13)$	$10^{**}(-11)$

Нетрудно заметить, что optuna может конкурировать с Nelder-Mead и классическим градиентным спуском, как минимум в плане стабильности, но вот до уровня Ньютоновских и Квазиньютоновских методов optuna не дотягивает даже близко.

Дополнительное задание 2

Нами было решено изучить возможности optuna на базе “Adam”, “RMSprop” и “Adagrad”

Список и значения гиперпараметров из третьей лабораторной работы :

```
"Adam": {"learning_rate":
        [0.01, 0.001, 0.0001, 0.00001],
        "beta_1":
        [0.9, 0.85, 0.8, 0.75, 0.5],
        "beta_2":
        [0.999, 0.999999, 0.95, 0.9],
        "amsgrad": [False, True]},

"RMSprop": {"learning_rate":
            [0.01, 0.001,
             0.0001, 0.00001],
            "rho":
            [0.9, 0.75, 0.5, 0.25, 0.1],
            "centered":
            [False, True]},

"Adagrad": {"learning_rate":
            [0.01, 0.001, 0.0001,
             0.00001],
            "initial_accumulator_value":
            [0.1, 0.05, 0.25]}}
```

Давайте вспомним результаты полученные во время выполнения третьей лабораторной работы(Epochs=64, Batch_size=256):

1) Adam

Best_loss: 2772358144
learning_rate: 0.01
beta_1: 0.5
beta_2: 0.999
amsgrad: True

2) RMSprop

Best_loss: 2686508032
learning_rate: 0.01
rho: 0.25
centered: True

3) Adagrad

Best_loss: 7386808832
learning_rate: 0.01
initial_accumulator_value: 0.25

Теперь давайте посмотрим на результаты полученные с помощью optuna(Epochs=64, Batch_size=25, n_trials=100):

1) Adam

Best_loss: 2717511936
learning_rate: 0.008871374108991116
beta_1: 0.8459746435885729
beta_2: 0.9532159743045338
amsgrad: False

2) RMSprop

Best_loss: 2612644352

learning_rate: 0.005950416848779665

rho: 0.2020137163048854

centered: True

3) Adagrad

Best_loss: 7030209536

learning_rate: 0.009961119760081604

initial_accumulator_value: 0.10358843630249222

Нетрудно заметить, что optuna действительно справилась со своей задачей, найдя более оптимальные гиперпараметры для каждого из трёх optimizers