Лабораторная работа №2

Продвинутые методы

https://github.com/GooddiLK/AllEDa

Пластинин Алексей M3237
t.me/plstnn

Малков Александр M3237
t.me/AlexM_37

Кинзябулатов Эдуард M3237
t.me/Eduard7000

Кулебакин Дмитрий M3237
t.me/SinDat_tg

Цель работы:

Сравнить эффективность работы различных методов поиска минимума в зависимости от вида функций.

Используемые методы:

- Градиентный спуск с постоянным шагом, экспоненциальным затуханием, условиями Армихо и Вольфе.
- Метод Ньютона с постоянным шагом и поиском шага по условию Вольфе.
- Newton-CG & BFGS из scipy.optimize.

Реализация BFGS

- Задается начальная точка x_0. Инициализируется начальная аппроксимация матрицы Гессе H_0. Вычисляется градиент функции в точке $x_0 : \nabla f(x_0)$
- На каждой итерации алгоритм вычисляет направление движения:
  $d_k = -H_k \nabla f(x_k)$
  Для поиска шага используется backtracking line search для нахождения подходящего шага α_k, удовлетворяющего условию Армихо:
  $f(x_k + \alpha_k d_k) \leq f(x_k) + c_1 \alpha_k \nabla f(x_k)^T d_k, (c_1 \in (0,1))$

  Обновление точки: $x_{k+1} = x_k + \alpha_k d_k$

  Обновление матрицы H_k:
  $H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T$
  $\left(s_k = x_{k+1} - x_k,\ \ y_k = \nabla f(x_{k+1}) - \nabla f(x_k),\ \ \rho_k = \frac{1}{y_k^T s_k}\right)$
- Результатом является последняя принятая точка x_k - приближение локального минимума функции

Реализация метода Ньютона Ньютон работает следующим образом - на каждой итерации он вычисляет значение, производную и гессиан в точке
Далее с помощью формулы вычисляем p $= -(\nabla^2 f(x_k))^{-1} \nabla f(x_k)$ а в направлении $-\nabla m(x_k)$ вычисляем минимум внутри доверительного региона далее находим пересечение линии от найденного минимума до p с границей реиона доверия - это и будет наш кандидат на x_{k+1}
но чтобы понять, насколько хорошо мы аппроксимировали нашей моделью f внутри доверительного региона считаем

$$\frac{\text{result} - f(x_{k+1})}{\text{result} - m(x_{k+1} - x_k)}$$

- это показатель того, насколько хорошо мы аппроксимировали функцию в delta-окрестности - нашем регионе доверия, где m - модель, аппроксимировация нашей функции до 2 члена ряда

Исследование:

Рассматриваемые функции:

- $x^2 + y^2$
- $3x^2 - 4xy + 10y^2$
- $(x^2 + y - 11)^2 + (x + y^2 - 7)^2$ - Функция Химмельблау
- $20 + (x^2 - 10\cos(2\pi x)) + (y^2 - 10\cos(2\pi y))$ - Функция Растригина

Начальная точка - (100, −200).

Лучшие значения гиперпараметров, подобранные с помощью optuna:

Константный шаг, Экспоненциальный, критерии Армихо и Вольфе

-	l_r	-	h_0	λ	-	α_0	q	ε	c_1	-	α_0	ε	c_2	c_1
F_1	0,41	-	0.49	1.6e-7	-	0.49	0.05	8.4e-7	0.07	-	1,00	5.8e-3	5.7e-4	2.1e-8
F_2	0.08	-	0.18	0.05	-	0.18	0.82	2.4e-6	6.2e-4	-	0.77	1.4e-10	5.7e-6	2.5e-6
F_3	Nan	-	Nan	Nan	-	0.05	0.57	2.9e-9	1.7e-3	-	1.93	1.5e-9	1.4e-4	4.5e-6
F_4	0.03	-	0.21	1.4e-3	-	0.51	0.09	1.2e-3	5.7e-4	-	4	2.7e-7	3.2e-4	1.8e-8

BFGS, Newton с критерием Вольфе

-	α_0	ε	c_1	q	-	α_0	ε	c_2	c_1	Δ	$\Delta_{\max}$	$\Delta_{\min}$	η	γ
F_1	1.02	0.01	0.04	0.27	-	0.06	3e-10	0.31	1.1e-6	5.7	0.57	0.07	0.01	2.1
F_2	0.98	1.6e-4	0.01	1.9e-5	-	0.04	3.6e-5	0.02	1.2e-6	9.2	0.5	0.03	2.1e-3	1.1
F_3	0.35	1.4e-6	3.2e-5	4.5e-3	-	2.1	8.6e-6	0.04	1.1e-3	0.20	0.95	0.07	0.02	1.16
F_4	4.66	2.1e-4	6.8e-4	0.47	-	3.80	4.7e-8	0.15	7.6e-5	0.05	0.53	0.02	2.3e-3	4.59

$\Delta_{\max}$ - trust_upper_bound - верхняя граница того, как хорошо предсказывает наша модель функции (если выше неё, значит можем расширить диапазон доверия)
$\Delta_{\min}$ - trust_lower_bound - минимальное значение достоверности модели (если ниже неё, то диапазон доверия уменьшается)

$\boldsymbol{\eta}$ - trust_no_trust_bound - минимальное значение достоверности модели для принятие результата как удовлетворяющего (если ниже, то приходится пересчитывать шаг с уменьшенным доверительным радиусом)

$\boldsymbol{\gamma}$ - trust_changing_multiply_value - множитель для изменения радиуса доверительной области

-	Point	Итерации	f	f'
Постоянный	(8.2e-8, −1.6e-7)	13	0	12
Экспоненциальный	(5.5e-9, −1.1e-8)	7	0	6
Армихо	(2.2e-10, −4.4e-10)	7	7	6
Вольфе	(-8.4e-10, 1.7e-9)	6	6	6
BFGS	(7.2e-9, −1.4e-8)	8	9	7
scipy BFGS	(-2.1e-14, 2.8e-14)	2	6	6
scipy Newton-CG	(0, 0)	2	2	2
Newton Вольфе	(0, 0)	2	2	2
Newton Эксп	(0, 0)	3	3	3
Newton Конст	(0, 0)	3	3	3
-	-	-	-	-
Постоянный	(-3.3e-8, 3.3e-7)	44	0	43
Экспоненциальный	(-3.0e-8, 1.4e-7)	32	0	31
Армихо	(9.8e-8, −3.7e-7)	69	341	68
Вольфе	(1.2e-7, 1.7e-8)	17	51	51
BFGS	(-4.0e-9, −1.4e-9)	10	12	9
scipy BFGS	(2.2e-16, −2.2e-16)	5	10	10
scipy Newton-CG	(0, 0)	7	7	7
Newton Вольфе	(0, 0)	2	2	2
Newton Эксп	(0, 0)	3	3	3
Newton Конст	(0, 0)	3	3	3
-	-	-	-	-
Постоянный	Nan	-	-	-
Экспоненциальный	Nan	-	-	-
Армихо	(-3.78, −3.28)	21	129	20
Вольфе	(3.58, −1.84)	35	278	278
BFGS	(60.4, 32.8)	4	10	3
scipy BFGS	(-2.80, 3.13)	34	44	44
scipy Newton-CG	(3.58, −1.84)	17	18	18
Newton Вольфе	(2.98, 2.04)	5000	6671	5000
Newton Эксп	(2.94, 2.04)	2886	4319	2886
Newton Конст	(-3.65, 5.04)	668	1013	668
-	-	-	-	-
Постоянный	(-4.17, −1.34)	5001	0	5000
Экспоненциальный	(2.5e-7, 2.5e-7)	2738	0	2737
Армихо	(-1.98, 3.98)	31	89	30
Вольфе	(-1.98, 5.97)	30	250	250
BFGS	(-2.98, 1.00)	12	51	11
scipy BFGS	(-21.9, 19.0)	12	22	22
scipy Newton-CG	(-5.96, −1.99)	12	30	30
Newton Вольфе	(3.98, 3.98)	3	4	3
Newton Эксп	(9.95, 7.96)	7	8	7
Newton Конст	(9.95, 7.96)	4	5	4

В Ньютоновских методах количество вызовов Гессиана равно кол-ву вызовов производной.

Выводы:

На **простых функциях** методы Ньютона работают лучше всего, достигая минимума за 1-2 итерации.
Градиентные методы тоже работают, но требуют больше итераций.

Для **функции Химмельблау**:
Градиентные методы с постоянным/экспоненциальным шагом не сошлись.
Методы с условиями Армихо и Вольфе нашли минимум с разной эффективностью.
Newton-CG и BFGS из scipy показали хорошие результаты.
Собственная реализация BFGS не справилась.

Для **функции Растригина**:
Градиентные методы часто застревают в локальных минимумах.
Методы Ньютона сходятся быстро, но не к глобальному минимуму.
Никто не справился с задачей. (Кроме экспоненциального шага, как так).