Национальный исследовательский университет ИТМО
Факультет информационных технологий и программирования
Прикладная математика и информатика

Методы оптимизации

Отчет по лабораторной работе №1

⟨Собрано 26 марта 2023 г.⟩

Работу выполнили:
Бактурин Савелий Филиппович M32331
Вереня Андрей Тарасович M32331
Сотников Максим Владимирович M32331

Преподаватель:
Свинцов Михаил

Задача 1

Постановка задачи

Реализуйте градиентный спуск с постоянным шагом (learning rate).

Решение

Поймем, изначальное, чего мы хотим добиться: мы хотели бы найти приближенный минимум на плоскости у заданной непрерывной функции f. Однако, при наивном решении этой задачи возникает проблема с производительностью нахождения $\operatorname{argmin} f$ за счет появления тех или иных накладных расходов на подсчет не целочисленных значения, а также всецело такого алгоритма, который бы с точность до некоторого изменения ε не «застрянет» в бесконечном поиске интересующей точки.

Для таких целей самым простым, но действующим метод является *градиентный спуск с постоянным шагом*. Введем обозначения, пусть λ – есть некоторая константа, порядка 10^{-3}, $x_i = \{x_i^0,\ x_i^1,\ \ldots,\ x_i^{n-1}\}$ – некоторая координата в n-мерном пространстве, p_i – наше текущее направление.

Теперь рассмотрим идею: оптимизацию нахождения необходимого минимум за k шагов мы будем осуществлять шаги в n-мерном пространстве в направлении, задаваемый как антиградиент функции f в точке, задаваемая предыдущем шагом, то есть

$$x_{i+1} = x_i - \lambda \cdot \nabla f(x_i),$$

где x_0 будет задаваться некоторым множеством $\mathrm{INIT} = \{x_0^0,\ x_0^1,\ \ldots,\ x_0^{n-1}\}$ – то есть точка, от которой мы собираемся двигаться.

В качестве промежуточного итога предоставим псевдо-алгоритм для решения этой задачи:

function $f(x)$:
    $/{*}implementation\ defined{*}/$

function $\nabla f(x)$:
    return $\left[f(x)\dfrac{\partial}{\partial x^0},\ \ f(x)\dfrac{\partial}{\partial x^1},\ \ \ldots,\ \ f(x)\dfrac{\partial}{\partial x^{n-1}}\right]$

function $gradient(f(x))$:
    $x_0 \leftarrow$ INIT
    $\lambda \leftarrow$ const
    $\forall i \in [1, k]$:
        $x_i \leftarrow x_{i-1} - \lambda \cdot \nabla f(x_{i-1})$

Задача 2

Постановка задачи

Реализуйте метод одномерного поиска (метод дихотомии, метод Фибоначчи, метод золотого сечения) и градиентный спуск на его основе.

Решение

Задача 3

Постановка задачи

Проанализируйте траекторию градиентного спуска на примере квадратичных функций. Для этого придумайте две-три квадратичные функции от двух переменных, на которых работа методов будет отличаться.

Решение

Задача 4

Постановка задачи

Для каждой функции:

(a) исследуйте сходимость градиентного спуска с постоянным шагом, сравните полученные результаты для выбранных функций;

(b) сравните эффективность градиентного спуска с использованием одномерного поиска с точки зрения количества вычислений минимизируемой функции и ее градиентов;

(c) исследуйте работу методов в зависимости от выбора начальной точки;

(d) исследуйте влияние нормализации (scaling) на сходимость на примере масштабирования осей плохо обусловленной функции;

(e) в каждом случае нарисуйте графики с линиями уровня и траекториями методов;

Решение

Задача 5

Постановка задачи

Реализуйте генератор случайных квадратичных функций n переменных с числом обусловленности k.

Решение

Для начала поймем, что такое *число обусловленности*. По своей сущности, это нечто, что может показать насколько измениться значение функции при небольшом изменении аргумента. Для нахождения такого числа и, в следствии, нахождения вектора чисел, которые будут являться коэффициентами квадратичной сгенерированной функции. Для решении такой задачи мы могли воспользоваться правильными методами такими как *теорема о сингулярном разложении*: возьмем некоторую матрицу

A, возьмем его после разложении образовавшийся диагональную матрицу, тогда его (матрицы A) число обусловленности будет равно отношению максимального по модулю и минимального по модулю собственных чисел выбранной матрицы. Проблемой столь мощного инструментария заключается в том, что генерация подобной матрицы наивным методом может занимать немереное время, ибо асимптотику спектрального разложения, которое и является основным в сингулярном, никто предугадать не может.

Тогда приходит идея менее безболезненная, но более радикальная: скажем, что наша матрица изначально была подана диагональной, а значит наша генерации функции сводится к вычислению n-чисел на диагонали матрицы.

Итак, пусть дано число обусловленности k, положим $\mathrm{MAX} = k \cdot \mathrm{MIN}$ – максимальное по модулю значение собственного числа матрицы, а в качестве минимального возьмем случайное число из ограниченного операционной системой диапазоном, например $[0, 2^{\log_2 X} - 1]$. Тогда, наконец, все остальные элементы следует брать из диапазона $[\mathrm{MIN} + 1, \mathrm{MIN} \cdot k)$.

В качестве промежуточного итога предоставим псевдо-алгоритм для решения этой задачи:

```
function random(l, r):
        return randomized Ret ∈ [l, ..., r)

function generate(n, k):
        MIN ← random(0, 2^64 − 1)
        MAX ← MIN · k
        q ← [MIN,  MAX,  x_0  ...,  x_{n−3}],  ∀x_i = 0
        ∀i ∈ [2, n]:
                q_i ← random(MIN + 1,  MAX)
```

Задача 6 и 7

Постановка задачи

Исследуйте зависимость числа итераций $T(n, k)$, необходимых градиентному спуску для сходимости в зависимости от размерности пространства $2 \leqslant n \leqslant 10^3$ и числа обусловленности оптимизируемой функции $1 \leqslant k \leqslant 10^3$.

Решение

Дополнительное задание

Постановка задачи

Реализуйте одномерный поиск с учетом условий Вольфе и исследуйте его эффективность. Сравните полученные результаты с реализованными ранее методами.

Решение