**Лабораторная работа №1**

**«Производная, градиент и градиентный спуск»**

В этой работе вам предстоит разобраться с понятием производной и градиента, а также реализовать алгоритм градиентного спуска и его вариации в виде программы на python. Перед началом работы необходимо получить пример программного кода с заданиями у преподавателя по web-ссылке на python-блокнот (ссылка предоставляется на портале edu.gubkin.ru).

Перед защитой результатов лабораторной работы необходимо загрузить результаты на портал в виде файла “.ipynb”.

**Задание 1.**

В этом задании необходимо определить, имеет ли функция производную в x = 0 (список функций см. по ссылке в блокноте).

**Задание 2.**

Посчитайте производную f(x)=x^x в точке x(0)=e. Ответ округлите до одного знака после запятой. Указание. Представьте функцию f(x) как e^g(x) для некоторой g.

**Задание 3.**

В этом задании вычислить производную f(x)=tg(x)⋅ln(cos(x2)+1), в точке x0=0. Ответ округлить до двух знаков после запятой.

**Задание 4.**

Необходимо написать python-функцию, которая в качестве аргумента принимает:

* числовую функцию f, у которой необходимо вычислить производную;
* число ε необходимо использовать в качестве "малого шага" для приближённого вычисления производной.

Функция должна в свою очередь возвращать числовую функцию f′, равную производной функции f. Шаблон функции см. в блокноте по ссылке.

**Задание 5.**

В этом задании Вы должны найти минимум функций с помощью градиентного спуска. Вам на вход подаются функция func, ее производная deriv (\*), а также начальная точка start, на выходе – точка локального минимума. Для вашего удобства заранее написана функция для отрисовки траектории градиентного спуска (см. пример кода по ссылке). В первой реализации градиентного спуска можете предполагать, что на вход подаются функции с единственным, глобальным минимумом.

**Ответьте на следующие вопросы:**

1. Каким образом контролировать условие остановки алгоритма?

2. Как правильно менять величину шага (learning rate) от итерации к итерации?

**Задание 6.**

В этом задании тестовые функции могут содержать несколько локальных минимумов. Вам нужно найти глобальный минимум у каждой. В общем случае такая задача невыполнима, но у вас будут одномерные функции и все самое

интересное будет сосредоточено в районе нуля. А именно, известно, что глобальный минимум лежит в пределах (low, high) (параметры алгоритма). Вам нужно модифицировать градиентный спуск, который вы написали в предыдущем задании, чтобы он работал и в таком случае. Сначала запустите градиентный спуск из прошлого пункта на тестах из ноутбука. Скорее всего, некоторые из них не пройдут.

**Вопрос:** как модифицировать алгоритм из предыдущего задания, чтобы получить верный результат?

Не забывайте вызывать callback(x, f(x)) на каждом шаге алгоритма!

**Задание 7.**

Вычислите градиент следующей функции:

ψ(x,y,z)=sin(xz)−y2z+ex

Запишите ответ в виде строки "(ψ′x)∗i+(ψ′y)∗j+(ψ′z)∗k", где вместо ψ′x, ψ′y, ψ′z подставьте вычисленные частные производные (компоненты градиента).

Можно пользоваться символами "+", "-", "\*", "/", "^", "sin", "cos", ")", "(", "e".

Примеры записи формул можно увидеть в ячейках текста задания.

**Задание 8.**

У вас есть только функция, которую Вам отдают в качестве аргумента. Вы должны найти её минимум. Вы будете искать глобальный минимум, у вас это должно получиться лишь потому, что предлагаются "хорошие" функции.

Минимум нужно искать на Ω=(−5,5)×(−5,5).

Попробуйте использовать следующие параметры:

* отклонение при вычислении производной ε=10^−10
* критерий остановки: близость к ответу δ=10^−10
* кол-во итераций 10^4
* длина шага градиентного спуска lr=1

**Вопросы:**

1. Как модифицировать алгоритм из задания 6, чтобы получить верный результат?

2. Как в вашей программной реализации алгоритма выбирается начальное значение и как контролируется условие остановки?