+FEDERAL STATE AUTONOMOUS EDUCATIONAL INSTITUTION

OF HIGHER EDUCATION

ITMO UNIVERSITY

**Report**

**on the practical task No. 3**

**“Optimization functions”**

Performed by

Chernobrovkin T.V. (412642)

Academic group J4133c

Accepted by

Gladilin P.E.

St. Petersburg

2023

# Goal

Construct and investigate the performance of optimization functions gradient descent with moments and Adam based on ordinary gradient descent.

# Problem

1. Develop yourself using numpy library:

a) python function for implementation of gradient descent (GD) with momentum algorithm for the function of two variables f(x,y).

b) python function for implementation of ADAM optimization algorithm for the function of two variables f(x,y).

2. Come up with a function of two variables of an arbitrary form and implement the search for its minimum using those implemented in p.1 functions - a) and b).

3. Illustrate the process of finding an extremum in the form of a graph of the dependence of the value of the difference between two successive approximations of the solution (x\_next – x\_prev) on the iteration number N or in the form of 2D-plane graph.

4. Write a conclusions about the accuracy of the implemented algorithms.

# Theory

Gradient descent, gradient descent method is a numerical method of finding the local minimum or maximum of a function by moving along a gradient, one of the main numerical methods of modern optimization.

It is actively used in computational mathematics not only for direct solution of optimization (minimization) problems, but also for problems that can be rewritten in the language of optimization (solution of nonlinear equations, search for equilibria, inverse problems, etc.). The gradient descent method can be used for optimization problems in infinite-dimensional spaces, for example, for numerical solution of optimal control problems.

A problem with gradient descent is that it can bounce around the search space on optimization problems that have large amounts of curvature or noisy gradients, and it can get stuck in flat spots in the search space that have no gradient.

Momentum is an extension to the gradient descent optimization algorithm that allows the search to build inertia in a direction in the search space and overcome the oscillations of noisy gradients and coast across flat spots of the search space.

Adam optimizer is the extended version of stochastic gradient descent which could be implemented in various deep learning applications such as computer vision and natural language processing in the future years. It is an optimization algorithm that can be an alternative for the stochastic gradient descent process. The name is derived from adaptive moment estimation. The optimizer is called Adam because uses estimations of the first and second moments of the gradient to adapt the learning rate for each weight of the neural network. Adam is proposed as the most efficient stochastic optimization which only requires first-order gradients where memory requirement is too little.

In Adam instead of adapting learning rates based on the average first moment as in RMSP, Adam makes use of the average of the second moments of the gradients. Adam. This algorithm calculates the exponential moving average of gradients and square gradients. And the parameters of β1 and β2 are used to control the decay rates of these moving averages.

# Materials and methods

In this task, all calculations were performed on the student's personal laptop in Google Colab. The work was performed in the Python programming language.

# Results

1. На основе градиентного спуска был написан оптимизатор градиентного спуска с моментами и оптимизатор Adam. Результаты представлены на рисунке 1 и 2 соответственно.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Figure 1 – Gradient desent with momentum

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Figure 2 – Adam optimizer

1. Была придумана функция двух переменных произвольного вида (рисунок 3). Её поверхность представлена на рисунке 4.

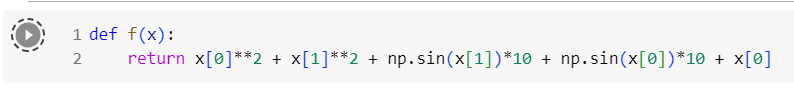


Figure 3 – A function of two variables

Изображение выглядит как диаграмма

Автоматически созданное описание

Figure 4 – Surface of our function

Полученные оптимизаторы были применены для нахождения минимума этой функции (рисунок 5).

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Figure 5 – Finding the minimum of a function

1. Процесс нахождения экстремума был изображён в виде графика зависимости величины разности двух последовательных приближений решения (x2 - x1 and y2 – y1) от номера итерации. Результаты для GD with momentum and Adam представлены на рисунке 6 и 7 соответственно.

Из полученных графиков видно, что с течением времени у GD with momentum растёт разница x2-x1 and y2-y1, однако примерно после 500 итерации алгоритма начинает сходиться.

У оптимизатора Adam не наблюдается таких провалов. Алгоритм достаточно долго находится на плато и примерно после 6000 начинает стремительно сходиться.

Изображение выглядит как диаграмма, График, линия, текст

Автоматически созданное описание

Figure 6 – The process of finding the minimum by gradient descent with momentum

Изображение выглядит как текст, снимок экрана, диаграмма, линия

Автоматически созданное описание

Figure 7 – The process of finding the minimum by Adam

1. Оба алгоритма работали при одинаковой точности 0.0001. Алгоритм GD with momentum сходится примерно в 4 раза быстрее чем Adam. При работе используемый мной код алгоритма Adam имеет следующую особенность – ему необходимо несколько операций для начала корректной работы. В свою очередь алгоритм GD with momentum исправно работает с условием сходимости |x2 - x1| > error and |y2 – y1| > error.

# Conclusion

При выполнении данного задания были построены функции оптимизации градиентного спуска с моментами и Адама на основе обычного градиентного спуска, а также исследованы их работа.

С их помощью был найден минимум нашей произвольной функции. По полученным результатам оптимизатор Adam работает в 4 раза медленнее.

# Appendix

GitHub link: