Optimization & Optimal Research
Practical Session

Abstract

The aim of this practical session is to implement several algorithms that we have seen during the class, study the convexity of various functions, find their global or local minima. We will also verify the influence of the learning rate for the gradient descent with a fixed step and see a convergence condition of this algorithm. You can use any programming language you want - I will try to help you but this should not be taken for granted.

Introduction (20 minutes)

Consider the following functions:

$$f_1(x,y) = x^2 + \frac{y^2}{20},$$
$$f_2(x,y) = \frac{x^2}{2} + \frac{y^2}{2},$$
$$f_3(x,y) = (1-x)^2 + 10(y-x^2)^2,$$
$$f_4(x,y) = \frac{x^2}{2} + x\cos(y).$$

For f_3, we will take $(x_0, y_0) = (-1, 1)$ as the initialization of our algorithm.

1. Compute the gradient of each function.
2. Which of the functions are convex? Why? Justify your answer using the arguments seen during the previous lessons.
3. Plot these functions.
4. What is the global minimum of each function?

We now want to solve the following optimization problem

$$\min_{(x,y)\in\mathbb{R}^2} f(x,y),$$

for each function f using the algorithms studied before.

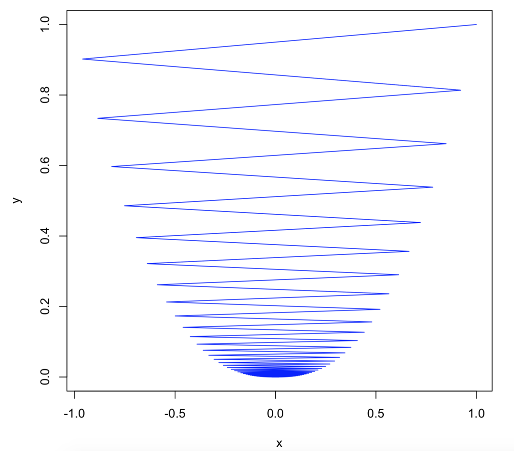
1 Gradient descent with a constant learning rate (40 minutes)

We recall that the gradient descent algorithm with constant learning rate $\eta > 0$ updates the weights at each iteration as follows:

$$u_{k+1} \leftarrow u_k - \eta \nabla f(u_k).$$

1. Define a function called *GradientDescent* implementing the gradient descent algorithm.

2. Test your function with different values of η for all functions (f_1, f_2, f_3, f_4). What do you notice? Represent the convergence of (x, y) in a graph resembling the following one:



3. Consider now function f_2. Test your algorithm with this function by setting η to 1.9 and 2.1. Give a condition on η so that the algorithm converges.

   We suppose that the function f is $\alpha-$elliptical and the gradient function ∇f is $L-$lipschitzien. We can show that this algorithm converges if we take η such that : $0 < \eta < \dfrac{2\alpha}{L^2}$.

   (a) Find the value of L such that :

   $$\|\nabla f(x_1, y_1) - \nabla f(x_2, y_2)\| \leq L\|(x_1, y_1) - (x_2, y_2)\|$$

   (b) Compute $\lambda_{\min}$, the smallest eigenvalue of f. Set $\alpha = \lambda_{\min}$ and conclude.

   (c) Try to do the same for the function f_1.

2 Gradient descent with optimal step (30 minutes)

For this exercise, the learning rate is not constant but determined by solving the problem :

$$\eta^{(k)} = \underset{eta>0}{Argmin}\ f(u_k - \eta \nabla f(u_k)).$$

1. Give an explicit expression of η for the f_1 and f_2.

2. Implement the algorithm with an adaptive learning rate.

3. Apply your modified algorithm to function f_1 and compare the result with the one obtained by the previous implementation.

4. Do the same for the function f_3.

3 Newton's Method (30 minutes)

The Newton's Method is solving Euler's equation

$$\nabla f(u) = 0.$$

An iteration of the Newton's algorithm is given by:

$$u_{k+1} \leftarrow u_k - (H_f(u))^{-1} \nabla f(u),$$

where H_j refers to the Hessian matrix of the function f.

Remark: It is possible to improve this method by combining it with a line search algorithm. To this end, we smay set:

$$u_{k+1} \leftarrow u_k - \eta (H_f(u))^{-1} \nabla f(u),$$

where η is a constant or a variable that satisfies the Wolfe's condition.

1. Compute the Hessian matrix for the functions f_1, f_3 and f_4.
2. Implement the Newton's method for the function f_3 and f_4 to find the global minimum.
3. For the function f_2 and/or f_3, compare the convergence of the three algorithms.
4. What are the advantage(s) and drawback(s) of this method?