Optimisation TD 1
Gradient descent and Newton method

C. Frindel

September 30, 2019

The goal of this exercise is to compare the two optimization methods seen in class : gradient descent (GD) and Newton's method (NM). Files `surface3d_demo.py` and `lines3d_demo.py` help you to represent a surface and a trajectory in 3D with the library matplotlib. These are useful tools for visualizing the functions in the exercsises.

1 Case 1 : Convex function from $\mathbb{R}^2$ to $\mathbb{R}$

We wish to know the minimum of the following function $f(x, y) = (x - y)^4 + 2x^2 + y^2 - x + 2y$.

1. Draw with `matplotlib` the function, locate the global minimum and local minima (if there are some). Work with the demonstration files provided. Justify what you're seeing.
2. Compute the gradient g of function f. What is its expression ?
3. Implement gradient descent method. Test the different stopping criteria seen in class and comment (start by the easiest one).
4. Run the algorithm with starting point $(x_0, y_0) = (1, 1)$ and a step of 0.09.
5. How many iterations does it take for the algorithm to converge ? What's the solution given by the algorithm, and gradient norm L_1 for this solution ?
6. Plot with `matplotlib` the trajectory $[x_0, x_1, \ldots, x_n]$ of the algorithm, and the function f. Test other values for the parameters of the algorithm (starting point, stopping conditions, step) and study the resulting trajectories.
7. Compute the Hessian of function f. What is its expression ?
8. Implement Newton's method. Use the same stopping conditions as for GD.
9. Run the algorithm for starting point $(x_0, y_0) = (1, 1)$.
10. How many iterations does it take for the algorithm to converge ? What's the solution given by the algorithm, and gradient norm L_1 for this solution ?
11. Plot with `matplotlib` the trajectory $[x_0, x_1, \ldots, x_n]$ of the algorithm, and the function f. Test other values for the parameters of the algorithm (starting point, stopping conditions, step) and study the resulting trajectories.
12. Compare results of the two methods.

2 Case 2 : Non-convex function from $\mathbb{R}^2$ to $\mathbb{R}$

We wish to find the minimum of $f(x, y) = x^2 - y^2$.

1. Draw with `matplotlib` the function, locate the global minimum and local minima (if there are some). Work with the demonstration files provided. Justify what you're seeing.
2. Run the algorithm of GD with the same stopping conditions as before. Take a step of 0.01. Test the starting points $(x_0, y_0) = (0, 0)$ and $(x_0, y_0) = (-5, 5)$. Comment the results.
3. Run the algorithm of NM with the same stopping conditions. Test the same starting points. Comment.
4. Compare the results of these two methods. Explain why they don't have the same behavior for the starting point $(x_0, y_0) = (-5, 5)$.

3 Case 3 : Non-convex function from $\mathbb{R}^2$ to $\mathbb{R}$

We wish to find the minimum of $f(x, y) = x^4 - x^3 - 20x^2 + x + 1 + y^4 - y^3 - 20y^2 + y + 1$.

1. Draw with `matplotlib` the function, locate the global minimum and local minima (if there are some). Work with the demonstration files provided. Justify what you're seeing.
2. Run the algorithm of GD with the same stopping conditions as before. Take a step of 0.01. Test the starting points $(x_0, y_0) = (3, 4)$, $(x_0, y_0) = (-3, -3)$ and $(x_0, y_0) = (-4, -3)$. Comment
3. Run the algorithm of NM with the same stopping conditions. Test the same starting points. Comment.
4. Explain why the solution returned by the two algorithms depends on choice of starting point. What should be the solution to return in all cases ?
5. Conclude on advantages and drawbacks of these methods, called "descent methods".