DS211, 2023 CDS, IISc

Numerical Optimization **Assignment-1** Due: 2/10/2023, 11:59pm

Submission Instructions

- The assignment is to be submitted through MS Teams as a single `jupyter notebook`.
- Maintain the naming convention of the notebook as follows.
  If your iisc email id is username@iisc.ac.in, then the name of the file should be username_assgn_1. For example, ksumanth_assgn_1 for my mail id of ksumanth@iisc.ac.in.
- Make sure that your cells in the Python Notebook are in the same order as the questions.
- Before submission, execute the command `Restart and Run all` from the Runtime/Kernel tab. Verify that there are no errors and you are getting the output as you expected.

Problem 1

Line search strategy

1. Consider the steepest descent method with exact line searches applied to the convex quadratic function $\frac{1}{2}x^TAx + b^Tx + c$. Mathematically, show that if the initial point is such that $x_0 - x^*$ is parallel to an eigenvector of A, then the steepest descent method will find the solution in one step.
2. Program the steepest descent and Newton algorithms using backtracking line search strategies. Compute the gradients and Hessian of a function using "Automatic Differentiation" (use either `Tensorflow`/`pyTorch` framework).
3. Use the above programs to minimize the classic Rosenbrock function (Function tolerance ¡ $1e-6$). Set the initial step length to 1. At each iteration store the step lengths used by each method and make plots. Show the step lengths taken and iterates as plots. Do these for a start point of search $x_0 = (1.2, 1.2)^T$ and then for the starting point $x_0 = (-1.2, 1.0)^T$.
4. Plot the convergence of the iterates and the objective function value. Evaluate the rate of convergence.
5. Call built-in functions for steepest descent and Newton's method, and show the results for the above. Compare and evaluate your program.
6. Use inbuilt Adam optimizer to minimize the Rosenbrock function with a learning rate of $1e-2$, keeping maximum iteration as 5000. Repeat the same with the learning rate of $1e-3$. comment on the convergence rates for two different learning rates. Is Adam optimizer doing better than the steepest descent method, justify your answer with a proper explanation.

Note: If you are using `TensorFlow` for Automatic Differentiation try to use `@tf.function` decorator to reduce your computation time.

Problem 2

Quasi-Newton and Conjugate gradient

1. Write a program that implements the BFGS method. Apply it to minimize Rosenbrock's function. Plot and show your results in as much detail as possible, highlighting the iterates, and convergence.

2. Show that for a quadratic function, with exact line search, the Polak-Reibiere formula, Hestenes-Stiefel formula, and Fletcher-Reeves formula are identical.

3. Implement the conjugate gradient method with Fletcher Reeves formula and use it to solve linear systems in which the Hessian is the Hilbert matrix, whose elements are $H_{i,j} = \frac{1}{i+j-1}$. Set the right-hand-side to $b = (1, 1, \cdots, 1)^T$ and the initial point to $x_0 = 0$. Try dimensions $n = 5, 8, 12, 20$ and report the number of iterations required to reduce the residual below $1e - 6$.