Duke University
ME 555. Numerical Optimization
Project Topics

1. Sequential Quadratic Programming (SQP) for equality constrained nonlinear problems.

   (a) Study the Newton's method with equality constraints from Chapter 10 in Convex Optimization by Boyd and Vanderbergue. No convergence analysis needed. Just the solution approach.

   (b) Study how to handle infeasible starting points.

   (c) Solve the equality constrained entropy maximization problem below using Algorithm 10.1 in Boyd and Vanderbergue. Solve the full KKT system at each iteration using a linear solver of your choice.

   $$\begin{aligned} \text{minimize} \quad & f(\boldsymbol{x}) = \sum_{i=1}^{n} x_i \log x_i \\ \text{subject to} \quad & A\boldsymbol{x} = \boldsymbol{b} \end{aligned}$$

   where $\boldsymbol{x} \in \mathbb{R}^n_{++}$, $A \in \mathbb{R}^{p \times n}$, $p < n$. Use $n = 1000$ and generate a matrix A randomly ensuring that it is full rank. To choose a feasible initial guess, randomly generate a vector $\hat{\boldsymbol{x}}$ with positive entries (e.g. in $(0, 1]$) and then set $\boldsymbol{b} = A\hat{\boldsymbol{x}}$. Solve this problem using: 1) A feasible start by choosing $\boldsymbol{x}_0 = \hat{\boldsymbol{x}}$ and 2) an infeasible start $\boldsymbol{x}_0 = \{1, 1, 1...\}$. Compare your answers. Report for each case the number of Newton iterations. Compare the performance and answerts of your implementation against Scipy optimization.

2. Solve the constrained entropy maximization problem in Topic 1 using the Augmented Lagrangian Approach with gradient projection discussed in class. Include simple bound constraints in the problem as $0 < l \leq x_i \leq u$. Use a trust region algorithm with CG-Steihaugh for the AL subproblem as discussed in class. You can implement the gradient projection method in Section 16.7 in Nocedal and Wright or as in the posted paper by Lin and Moré. Use $n \geq 10$ for the project.

3. Implement a Trust Region Newton-CG algorithm to train a neural network. Use a matrix-free approach to implement the action of the hessian on a vector through forward and back propagation steps. See Chapter 5 in Pattern Recognition and Machine Learning by C. Bishop for an explanation on how to derive and implement the action of the hessian of a neural network on a vector using a matrix-free approach. Demonstrate your implementation in a problem with at least 1000 weights and 10,000 data samples. Compare the performance of the implementation against stochastic gradient descent. You can use existing software for the stochastic gradient descent portion.

4. Propose your own project. For this, submit a one-page document describing the problem and the methods you will implement. This project must include topics and algorithms from the optimization literature (discussed or not discussed in class). The instructor needs to review and approve your proposed project.