Advanced Numerical Linear Algebra Homework 4
AMSC/CMSC 763 Due November 27, 2019

Problem 1. The paper

> C. F. Higham and D. J. Higham, "Deep learning: An introduction for applied mathematicians," arXiv:1801.05894v1, 2018.

is an introduction to machine learning and that contains a detailed example in which an artificial neural network differentiates data in a two-dimensional domain into two categories (the "oil" and "no-oil" categories discussed in class).

Write a program that implements a neural net with stochastic gradient descent and tests it on the data given in Listing 6.1 of this paper. For your program, you can borrow from the one given in the paper, `netbp`, but your program should be more general than `netbp` to allow a variable number of hidden layers as well as variable numbers of neurons in each layer.

Explore the performance of your network for several (say ~ 3) different numbers of layers and for one such choice, several numbers of neurons. Similarly, explore the impact of the learning rate η. Plot the results in images like Figure 4. Explore what happens to the output when you add a data point as in Figure 5 of the paper.

Problem 2. Suppose you have a linear system of equation $A(\boldsymbol{\xi})u(\boldsymbol{\xi}) = f$ where $\boldsymbol{\xi} = [\xi_1, \xi_2, \ldots, \xi_m]^T$ is a vector of parameters and

$$A(\boldsymbol{\xi}) = A_0 + \xi_1 A_1 + \xi_2 A_2 + \cdots + \xi_m A_m.$$

Here, the individual matrices $\{A_j\}_{j=0}^m$, each of order N, are given, but the parameter values are not generally known, and the solution $u(\boldsymbol{\xi})$ will also depend on the values of these parameters. We are interested in computing statistical properties of the parameter-dependent solution. This exercise will focus on the mean. We will also assume that the individual parameters are independent and uniformly distributed in $[-1, 1]$, so that we want to understand

$$\bar{u} = \int_{[-1,1]^m} u(\boldsymbol{\xi}) d\boldsymbol{\xi}.$$

a. One way to estimate $\bar{u}$ is by Monte Carlo simulation to compute a sample mean $\bar{u}_s$:

```
for j = 1 : n_ξ
   sample the m-vector ξ, giving ξ^(j)
   solve A(ξ^(j))u(ξ^(j)) = f
end
compute ū_s = (1/n_ξ) Σ_{j=1}^{n_ξ} u(ξ^(j))
```

What is the computational complexity of this procedure? State in particular any assumption you make concerning its dependence on N.

b. A strategy designed to produce an approximation to $\bar{u}$ at lower cost is as follows.

1. Generate a number of sample solutions $\{u(\boldsymbol{\xi}^{(j)}), j = 1, \ldots, n_s\}$, which will be referred to as *snapshots*.
2. Find a matrix V with $n_r \leq n_s$ columns whose range space represents a good approximation to $span\{u(\boldsymbol{\xi}^{(1)}), \ldots, u(\boldsymbol{\xi}^{(n_s)})\}$.
3. For any $\boldsymbol{\xi}$ different from those used to generate the snapshots, compute $y(\boldsymbol{\xi})$ such that $\hat{u}(\boldsymbol{\xi}) = Vy(\boldsymbol{\xi})$ whose residual $f - A(\boldsymbol{\xi})\hat{u}(\boldsymbol{\xi})$ is orthogonal to $range(V)$.

The idea behind this approach is that the collection of sample solutions from step (1) should characterize the solution set for all possible values of $\boldsymbol{\xi}$, and that there might in fact be more of them then is necessary. In this case, step (2) reduces the size of the needed set to n_r, which we would like to be small. This methodology has a name, *reduced-order models*.

Here are some questions about this:

b-Qi. Given the snapshots from item (1), how would you compute V in step (2)?

b-Qii. Each solution $y(\boldsymbol{\xi})$ can be computed at cost depending on n_r but not N, provided certain quantities are precomputed once and saved. Describe a way to accomplish this and specify the computational costs after the preliminary computations are done.

b-Qiii. This procedure can be repeated $n_{\boldsymbol{\xi}}$ times, giving $n_{\boldsymbol{\xi}}$ solutions $y(\boldsymbol{\xi})$. These solutions can be then used to construct an estimate $\hat{\bar{u}}_s$for the sample mean. Show that $\hat{\bar{u}}_s(\boldsymbol{\xi})$ can be computed with complexity $O((N + n_{\boldsymbol{\xi}})n_r)$.

c. Returning to step (1) above, another way to use the sample solutions produced there is to develop a machine learning (ML) model using these results. Describe what such an ML construction would entail, how it would be used, and describe an advantage it might have over the method developed in the rest of part (b).