Homework 1
APPM 4720/5720 Scientific Machine Learning, Fall 2024

Due date: Friday, Aug. 30 '24, before midnight, via Gradescope

Instructor: Prof. Becker
Revision date: 8/23/2024

Theme: Basic neural nets and review of numerical analysis

Instructions Collaboration with your fellow students is OK and in fact recommended, although direct copying is not allowed. The internet **is allowed for basic tasks** (e.g., looking up definitions on wikipedia, looking at documentation, looking at basic tutorials) but it is not permissible to search for solutions to the exact problem or to *post* requests for help on forums such as `http://math.stackexchange.com/`.

Problem 1: Form a group of between 3 and 5 students, including at least one student enrolled in 4720 and at least one student enrolled in 5720. $\boxed{\text{Write down the names of your group members.}}$

Problem 2: Create and train a neural net on the 50k training data in CIFAR10, and achieve at least 50% average classification on the 10k standard testing points. $\boxed{\text{Turn in a PDF of your source code}}$, **nicely formatted** (no screenshots!). *The README.md file at* `https://github.com/cu-applied-math/SciML-Class/tree/main/Homeworks` *has suggestions on how to export code to PDF nicely. You can find the CIFAR10 dataset all over the internet, including nice repositories like HuggingFace (which is worthwhile learning how to use). You can use any programming language, and you are allowed to follow demos/tutorials you find on the internet (in fact, I recommend you look online to find a good architecture), but be aware that in later homeworks we will do more with this code, so make sure you understand it.*

Problem 3: Let $d = 500$ and $\boldsymbol{x}_0 \in \mathbb{R}^d$ be the all ones vector. For any $n \in \mathbb{N}$, let $G \in \mathbb{R}^{n \times d}$ be a matrix where each entry is drawn i.i.d. from a standard normal distribution, and similarly let $\boldsymbol{z} \in \mathbb{R}^n$ also have i.i.d. standard normal entries.

Define $A = G + (1 - 10^{-7})\widetilde{G}$ where the j^{th} column of $\widetilde{G}$ is the $(j-1)^{\text{th}}$ column of G, done circularly (so the 1st column of $\widetilde{G}$ is the last column of G). *See* `np.roll` *in Python or* `circshift` *in Julia and Matlab*

Let $\sigma = 0.1$ and define $\boldsymbol{b} = A\boldsymbol{x}_0 + \sigma \boldsymbol{z}$.

a) Write code that returns the least-squares solution $\hat{\boldsymbol{x}} \stackrel{\text{def}}{=} \operatorname{argmin}_{\boldsymbol{x}} \left(f(\boldsymbol{x}) \stackrel{\text{def}}{=} \|A\boldsymbol{x} - \boldsymbol{b}\|_2 \right)$, and turn in a PDF of your code, and write a few sentences about how you know that your code is correct (test it on $n = 1000$). *You are allowed to use external linear algebra libraries; you don't have to do this by hand!* Your code should solve the problem in $\mathcal{O}(n)$ time (to be specific, $\mathcal{O}(nd^2)$).

b) What are the values of $f(\hat{\boldsymbol{x}})$ and $f(\boldsymbol{x}_0)$? (as before, use $n = 1000$).

c) Provide numerical evidence that your solver is indeed $\mathcal{O}(n)$. *This is the main part of the problem; think carefully about how to demonstrate this.*

Problem 4: Finite differences. Let $f(x) = \sin(x)$ and $x_0 = 2$. Approximate the derivative of f at x_0 using a finite difference scheme with stepsize h (e.g., the simplest such scheme is forward differences, $f'(x_0) \approx h^{-1}\left(f(x_0 + h) - f(x_0)\right)$). $\boxed{\text{Turn in a PDF of your code}}$ and a

$\boxed{\text{plot of the error for different values of } h}$, and $\boxed{\text{comment on whether you can achieve } 10^{-14} \text{ error.}}$
To calculate the error, you can use the fact that f' is known. When choosing values of h, choose reasonable values, and also decide how to scale your plot (i.e., make some axes logarithmic?).