Homework 1: PyTorch Checkup

ENM5320 Spring 2025, UPenn

Due date: January 29th by midnight (11:59pm)

This is an **individual assignment**; you are encouraged to use any/all resources but you must attribute any resources used explicitly (collaborators/textbooks/journal papers/ChatGPT). Submit a hand-written or Latexed report in Canvas, alongside Jupyter notebooks, figures, or any other supporting files needed. Code should be documented sufficiently that we can run the code without modification and reproduce your results. Late submissions will follow the guidance outlined in the course syllabus.

This first assignment is meant to be easy for those who are familiar with PyTorch. It is OK if you are not, but you must reach out quickly so that we can catch you up or you will not be able to keep up with the course. We will very happily step you through how to complete the assignment. We will reuse aspects of the code generated here later in the course.

Instructions:

On the course github, you can find a Jupyter notebook providing step-by-step examples of how to use pytorch for basic learning tasks (`ENM5320 > Code > Lecture01.ipynb`, Link). You can click the link at the top to directly load into Colab. Step through the notebook and confirm that you can follow the syntax; you only need to [shift+Enter] through the whole notebook to execute the whole thing. Be sure to use Gemini to ask for clarification of any specific commands or syntax that are unclear. **It is your responsibility to quickly come to office hours early if you need extra help getting started!**

1. **Derive the least squares solve.** Here I will guide you through the derivation of the least squares solve used in the notebook. This is to confirm that you are comfortable with index/vector notation and differentiation at the level that we will use it in the class.

   Assume $u(x) = \sum_{\alpha=1}^{N_{basis}} c_\alpha P_\alpha(x)$ for a given basis $\mathbf{P}(x)$.

   - Show that the mean square error

$$\mathcal{L} = \frac{1}{N_{data}} ||u - u_{data}||^2$$

where $||f|| = \sum_{d=1}^{N_{data}} f_d^2$
can be written as

$$\mathcal{L} = \frac{1}{N_{data}}(c^\intercal P - u_d)^\intercal(c^\intercal P - u_d)$$

where $c \in \mathbb{R}^{N_{basis}}$, $P \in \mathbb{R}^{N_{basis} \times N_{data}}$, and $u_d \in \mathbb{R}^{N_{data}}$.

- Take the partial of $\mathcal{L}$ with respect to c_α (the α^{th} element of the vector c) to obtain N_{basis} equations that c must satisfy at a minimizer of the loss.
- Show those equations can be written in the following form, known as the system of *normal equations*.

$$P^\intercal Pc = P^\intercal u_d$$

2. **Run jupyter notebook with a different dataset.** The previously linked jupyter notebook shows at the end how to use a multilayer perceptron to solve for a basis that will fit a set of four functions well. As the course progresses, you will develop similar models that embed neural networks within a standard scientific computing task. To start in this assignment, we'd like to confirm that you are able to understand a model of that type which is given to you, and train it with a different dataset.

   - First, recall the definition of an *indicator function* or *characteristic function*:

$$\mathbb{1}_A(x) = \begin{cases} 1, & \text{if } x \in A \\ 0, & \text{otherwise} \end{cases}$$

   - Generate code to evaluate N different functions $f_i(x) = \mathbb{1}_{[\frac{i}{N+1}, \frac{i+1}{N+1}]}$ evaluated on a uniform grid of M points.
   - For $N = 4$. $M = 50$, swap f_i in the for the four function used in the example for the notebook.
   - Play with the width, depth, activation functions and learning rate until you are able to learn a good basis that fits all four functions well. This may be sensitive to the initialization of the neural network, so you may need to try a few times for a given set of parameters to get a good fit.
   - Generate plots of your basis, as well as of the least squares fit of your four functions to the basis. Comment on any thoughts you have about how this case is different from the four functions I provided in the tutorial.