ECE655 (Advanced GPU Programming and Deep Learning) Project1 (10 points)

Deadline: Sep 12, 2025 11:59PM. 2 point penalty per each late day

- In this project, use the code provided in the lecture 2 notes and generate results. The goal is to gain an intuitive understanding of the concepts introduced in Lecture 2.
- Document your results in a report, written in LaTeX (Overleaf). A nice LaTeX example file is included.
- Your computer must have the full Anaconda package to complete this project.
- Submit your report's entire Overleaf directory and the compiled PDF report file under
- **Project 1 Submission Area**, *as a single ZIP file*, which must include:
  - For your Python code: all of your source files, .py
  - For your Report: all of your source files, .the entire directory, also the PDF

PROJECT 1 DESCRIPTION:

Go online and download a simple dataset that has 100s of (x,y) data points.
I don't care what the data is about (solar irradiation, temperatures, noise measurements, whatever it is). The only requirement is that the relationship between x and y should be linear, or close enough. The data cannot be synthetic; it must be the real measurements of something. Use your common sense to look at the data points to see if they look linear.

PART A (1 point):

- Determine model parameters in your linear model. Your data points are y, x, and two model weight parameters are b, w. Of course, y and x will depend on the dataset you find.
- Using the linear regression package, find the best fit; these are your golden values. In other words, in PART A, you are using a pre-existing linear regression package, rather than finding the solution yourself through an iterative program.
- **PLOT 1:** Scatter plot data points and super-impose the best model fit (with the golden parameter values; title this plot **golden prediction**).

PART B (2 points):

- **PLOT 2:** Using MSE loss, plot the loss surface for different model parameter values.
- Choose 100 discrete values for each parameter, much like the book, centered around the golden parameter values. So, your loss plot will have 10000 data points. Use the code from Lecture 1 and Lecture 2.

PART C (3 points):

- In this part, you are trying to predict the model parameters using a Naïve algorithm as follows:
- Create a list named `Iterations`. It will record tuples with b, w, MSE.
- At the beginning, the list is empty.
- Choose a **random** b, w value pair for your model parameters. Calculate the MSE loss for these parameters. Append it to the `Iterations` list.
- This is your starting point. You are trying to beat this MSE in the consecutive random picks.
- Now choose another **random** b, w and calculate the MSE using them. One of two things could happen:
    - If this MSE is better than what you had before, append it to `Iterations`. This is a **hit**.
    - If this is worse than the previous one, do not append. This is a **miss**.
- Repeat this for a total of 1000 random b, w pairs.
- At the end of 1000 trials, you should have a bunch of tuples in your `Iterations` list that have MSE values that get progressively better. The last one is your final (and best) prediction.

PART D (2 points):

- **PLOT 3**: Plot the loss surface and super-impose the points you kept in your `Iterations` list. Add a red dot for the golden prediction (from PART A).
- Your initial prediction should be blue, all consecutive predictions should be black, and your final prediction should be blue, all the while the red dot is the golden prediction. You are welcome to try other coloring schemes to make it more visually-appealing.
- This will allow you to visualize how your picks progressively reduced the MSE loss.

PART E (2 points):

- All of the plots and code should be included in your report.
- It is very easy to save the plots from Python, although, you are welcome to use the SnippingTool to save figures as PNG or JPG. These pictures are very easy to include in LaTeX.
- Comment in your report: How does this algorithm compare to Gradient Descent? How close did you get to the golden prediction?
- Make sure to include how many hits and misses you had under **Conclusions** of your report.
- DO NOT plot, however, run the program and determine how many misses you get for 2000, 5000, and 10000 random picks. Compare it to 1000 picks this project requires.