Project-9 (Bridging Civil Engineering and ML)

Problem 1: Linear Regression

This question is to build a linear regression model from scratch (using any programming language, preferably python)

You need to apply your model to predict rainfall intensity from other given features For approaching this regression problem, you will implement linear regression model and apply it to the 'Rain_data' dataset. The dataset consists of 501 entries. Each entry has 3 input features and the output would be Rainfall intensity. The dataset "Rain_data.csv" is provided along with this assignment. This is a coding assignment. You can write suitable functions or code in python language preferably to work out the questions provided in this assignment.

Your task is to construct a Linear regression model to predict the rainfall intensity using the input features provided in this dataset. You will also have to write pseudo code for each function. Use of Linear Regression libraries/packages is prohibited for this part. You should follow the steps mentioned below:

a) Read the dataset Rain_data.csv. Define a function which Normalizes the input features to have mean 0 and standard deviation 1 for each feature. Split the dataset into training and validation in the ratio of 70:30.

b) Plot data - Define a function that plot the relation between rainfall intensity and different features.

c) Prediction function- Define a function that can predict the rainfall intensity using input features and weights.

d) Cost function (a.k.a. loss function)- Define the cost function using the root mean square error values. The function usually takes as input the features, corresponding labels, and the weights.

e) Weight update function- define the function for updating the model weights using gradient descent algorithm using features, labels, weights, and learning rate as input. You might need to use the prediction function defined in Part b.

f) Training- Define the linear regression function using the functions you defined in the earlier steps. Create other functions as required. Train the model using all input features available in the Rain_data dataset choosing a suitable learning rate. Plot cost function using the gradient descent against the number of iteration.

Model evaluation- Report the accuracy observed for the training and test data set.

The dataset used in this assignment is adapted from the National Centres for Environmental Prediction-Department of Energy (NCEP-DOE) Atmospheric Model Intercomparison Project (AMIP)-II Reanalysis (Reanalysis-2) dataset.

Problem 2: OpenCV Image Conversion

Description: You are provided with a CSV file containing 785 columns. The first column represents the label, and the remaining 784 columns contain pixel values. Your task is to convert the 784 pixel values into a 28x28 image using the OpenCV library and save each image with the corresponding label as the filename. Additionally, you should create a directory named "result" to store all the images, and create a zip file for convenient upload.

Instructions:

1. Load the CSV File: Begin by loading the CSV file into your program. You can use libraries such as Pandas or built-in file reading functions, depending on your chosen programming language.
2. Extract the Label and Pixel Values: Separate the label column from the pixel columns. Store the labels and pixel values in separate variables for further processing.
3. Reshape the Pixel Values: Since the pixel values are originally in a flattened format, reshape the array into a 28x28 matrix. This will allow you to visualize the image.
4. Create Images: Use the reshaped pixel values to create grayscale images. You can utilize OpenCV functions to create and manipulate images.
5. Save Images: Save each image with the corresponding label as the filename. Make sure to choose a suitable file format, such as JPEG or PNG, depending on your requirements.
6. Repeat for all Data Points: Iterate through all the data points in the CSV file and repeat the image conversion and saving process for each data point.
7. Verify Results: Verify that the images have been created and saved correctly. You can manually inspect a few images to ensure they match the corresponding labels.

Note: Make sure the programming language you choose supports the OpenCV library and its image processing functions.

Week-3
Assignment