DLOPS
DLOps Assignment 1 [100 Marks]
Deadline: 15/03/2024, 23:59:59

Programming instructions:

1. Programming language: Python
2. Use of PyTorch is compulsory. Marks shall not be given for TensorFlow implementation.

Reporting instructions:

1. Please submit all your working codes as .py **AND** .ipynb files/colab notebook links. The file should be named appropriately with your complete roll number XYZ **(e.g.: "XYZ_DLOps_Assignment1_Q<questionNo..ipynb"),** with your name, roll number, and question number.
2. A single report (PDF) file containing all relevant information, including data pre-processing, observations, results, and analysis, should be submitted. Make sure colab links are mentioned at the top of each report.
3. Separately attach the files (.py, .ipynb/colab link, .pdf) for submission. Do not zip them together.
4. Do not put snapshots of code in the report.
5. The report should be detailed and clearly explain every step you have followed. All the intermediate outputs and their inferences should be present in the report. The PDF file should be named appropriately with your complete roll number XYZ **(e.g.: "XYZ_DLOps_Assignment1.pdf").** Ensure your name and roll number are mentioned inside the report as well.
6. Mention any resources/articles/GitHub links that you may have used as a reference to solve any question of the assignment in the references section of the report.

General instructions:

1. **FOLLOW** the above reporting instructions; otherwise, your assignment will NOT BE EVALUATED.
2. **DO NOT** plagiarize from the internet or your peers. The institute's plagiarism policy will be strictly enforced.
3. The assignment will be evaluated out of 50% of the total marks; if a report is not submitted.
4. We recommend using Google Colab with GPU runtimes for this assignment.

Question 1: Implementing and training a Multi-Layer Perceptron (MLP) and analyzing the model using Tensorboard [25 marks]

- Implement an MLP to classify the IRIS dataset. The MLP should have 2 hidden layers, one with 5 neurons and one with 7 neurons. The input layer should have 4 neurons and the output layer should have 3 neurons.
- Construct a custom dataset for the IRIS dataset with appropriate preprocessing operations and implement a 10-fold cross-validation strategy for evaluating the model and comparing the performance metrics (recall, precision, accuracy) across folds.
- Utilize PyTorch's Tensorboard integration to plot precision-recall curve, training loss, and accuracy curves for a given classification task and generate t-SNE plots to visualize the feature representations learned by the MLP on the given dataset. Interpret the clusters formed and their relevance to the task at hand in the report.

Question 2: Implementing and training Convolutional Autoencoders [3 x 25 = 75 marks]

Build convolutional autoencoders (CAE) for the following three image transformation tasks with the Hymenoptera dataset. **a) an image in L*a*b* colorspace to the image in RGB colorspace, b) RGB image to negative image, and c) RGB Image to RGB horizontal flipped image.** Do the following for each of the three transformations:

- The spatial resolution of the input and output images must be 224 x 224.
- Choose a suitable CAE architecture with the convolution, batch norm, and leaky ReLU operations in the encoder block and transposed convolution, batch norm, and ReLU operations in the decoder block.
- Construct the custom dataset and apply appropriate preprocessing operations to load the data for feeding them into the CAE.
- Visualize the resulting images to evaluate the performance of the autoencoder and display the MSE, RMSE, SSIM, and PSNR.
- Generate t-SNE and PCA plots to visualize the feature representations (i.e., the output from the last convolution layer of the encoder) learned by the autoencoder on the given dataset.
- Train an MLP classifier using the features learned by an autoencoder. Utilize PyTorch's Tensorboard integration to plot precision-recall curves, training loss, and accuracy curves for the given classification task.