Mini-Project 3: Machine Learning

CP322-A, Fall-2023, Wilfrid Laurier University

General information

- **Due on 10-Dec-2023 at 11:30 pm**. Late work will be automatically subject to a 20% penalty and can be submitted up to 5 days after the deadline. No submissions will be accepted after these 5 days.
- All members of a group will receive the same grade unless no collaboration has been done by individuals.
- You will submit your assignment on MyLS as a group. The group leader is responsible for the assignment submission. Other group members should make sure that their leader has submitted the assignment on time before the deadline
- You must submit two separate files to MyLS (**using the exact filenames and file types outlined below**):
    - **CP322-A3-GroupID-code.zip**: Your data processing, classification, and evaluation code (.py and .ipynb files).
    - **CP322-A3-GroupID-writeup.pdf**: Your (max 5-page) project write-up as a pdf.
- Except where explicitly noted, you are free to use any Python library or utility for this project.
- Questions related to this assignment should be forwarded to your TA, MD Shahed Hossen, hoss1130@mylaurier.ca.
- To collaborate among group members, we recommend using **Overleaf** for writing your report and **Google Colab** for coding and running the experiments. The latter also gives access to the required computational resources. Both platforms enable remote collaborations. However, you are free to use whatever you prefer for your write-up and coding.
- Follow the same instructions as the previous assignment for both the write-up and evaluation.

Problem definition

In this mini-project, you need to develop models to classify image data. You will use **the CIFAR-10 dataset** with the default test and train partitions. You can use **'torchvision.datasets.CIFAR10'** and **'torch.utils.data.DataLoader'** to load the data or use the details provided on the website for more information. Apply and compare the performance of the following models on this dataset:

- Artificial Neural Network (ANN): implement this from what you learned in the class. Your implementation should include the backpropagation and the mini-batch gradient descent algorithm used. You are encouraged to change the activation function (e.g., use ReLU), increase the number of layers, and play with the number of units per layer.
- Convolutional Neural Network (CNN): you may find a coding tutorial and use it for your coding on your dataset for this model. However, you need to add comments for each line, and each function, to make sure you understand it completely.

Compare and report the test and train performance of the above two models as a function of training epochs. Optionally, you may also compare various models based on the total number of parameters, as well as the choice of hyper-parameters such as the number of layers, number of units or channels, and the activation function. You are free to use any Python libraries you like to extract features and preprocess the data, evaluate your model, tune the hyper-parameters, etc.