

MiniProject 3: Classification of Image Data

COMP 551 (001/002), Winter 2020, McGill University

General information

- **Due on April 14th at 11:59pm.** Can be submitted late until April 19th at 11:59pm with a 20% penalty.
- To be completed in groups of three, all members of a group will receive the same grade. You can work with new group members or same members that you have collaborated before with, whichever works best.
- To be submitted through MyCourses as a group. You need to register your new group on MyCourses and any group member can submit the deliverables, which are the following two files:
 1. **code.zip:** Your data processing, classification and evaluation code (.py and .ipynb files).
 2. **writeup.pdf:** Your (max 5-page) project write-up as a pdf (details below).
- Main TA: Arnab Kumar Mondal, arnab.mondal@mail.mcgill.ca
- We recommend to use **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.
- Follow the same instructions as the previous assignment for both the write-up and evaluation. The only difference is that we don't have a bonus for the best performance or a penalty for the test-to-train leakage¹.

Problem definition

In this mini-project, we will develop models to classify image data. We 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, see the tutorial below for more information. Apply and compare the performance of following models on this dataset:

- Multilayer perceptron: implement this from scratch based on the code available in the slides. Your implementation should include the backpropagation and the mini-batch gradient descent algorithm used (e.g., SGD). You are encouraged to change the activation function (e.g., use ReLU), and increase the number of layers, and play with the number of units per layer².
- Convolutional Neural Network: start with this tutorial, which provides a working code on the same data for this model. You can reuse this code, 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 choice of hyper-parameters such as the number of layers, number of units or channels, and the activation function². You are free to use any Python libraries you like to extract features and preprocess the data, evaluate your model, and to tune the hyper-parameters, etc. The only restriction is on the implementation of MLP and its optimization.

¹ You still need to avoid this to have a correct evaluation and get the correctness marks for evaluation.

² Suggestions to get the originality / creativity points. For detailed breakdown refer back to the last assignment.