COMP5212: Machine Learning **Fall 2022**

Programming Homework 2: Due Wednesday Nov. 9, 11:59 PM

Instructions:
In this homework you will practice building a multi-class image classification pipeline to classify daily life images (CIFAR10), based on the modern neural networks. Please keep the batch size untouched. The goals of this homework are as follows:

- understand how to use Pytorch to build multi-class classifiers.
- understand the mechanism of convolution in image classification.
- learn the power of non-linearity in modern neural networks.
- implement and apply a fully-connected multi-class image classifier.
- implement and apply a Convolutional Neural Networks (CNN) classifier.

Problem Description.

In this homework, you are asked to implement fully-connected (MLP) and Convolutional Neural Networks (CNN) image classifier on CIFAR10 dataset. In this task, you only need to perform multi-class classification. Details of these models could be found in lecture 14 page 3. We provide a skeleton code for data loading and iterations of training data. You are asked to implement the rest of training in Pytorch code. Detailed submission requirements are written in the final section.

Resources

You can follow the setup instructions at `https://pytorch.org/get-started/locally/`.
A useful tutorial on learning building CNN at `https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html`.
Convolutional functions could be found here: `https://pytorch.org/docs/stable/nn.html#convolution-functions`.

Data

We use CIFAR10 classification dataset. Pytorch/torchvision has provide a useful dataloader to automatically download and load the data into batches. We have written the data loader for you as follow. You can find it in the attached file.

Submitting your assignment

Upload a PDF report using LaTeX containing your experiment results to Canvas (remember to include your name and ID number in the report). Some requirements for your report:

- Limit the number of pages to 3.
- You are asked to implement a 7 layers fully-connected neural networks with ReLU activation function. The model accuracy should be around 50% percent.
- You are asked to implement a 7 layers convolutional neural networks, 4 convolutional layers and 3 fully-connected layers, with ReLu activation function. The input dimension of 1st fully-connected layer must be 4096. The model accuracy should be around 85% percent.
- Describe your 2 model structures including in channels, out channels, stride, kernel size, padding for CNN layer; input dim, out dim for fully connected layer.

- For each of the model, report the $(\sum_{b=1}^{B} \sum_{d=1}^{D_b} \frac{loss(label_{b,d}, f_b(data_{b,d})}{D_b})$/B for each training epoch, where B is the total number of batches, f_b is the model after updated by b-th batch and D_b is the number of data points in b-th batch. An epoch is defined as one iteration of all dataset. Essentially, during a training epoch, you record down the average training loss of that batch after you update the model, and then report the average of all such batch-averaged losses after one iteration of whole dataset. You could plot the results as a figure our simply list down. Please at least report 10 epochs.
- Report the final testing accuracy of trained model.
- Please compare results for 2 models (MLP and CNN)).
- Try neural network without non-linear activation functions and discuss your findings.

Also, upload your code in a zip file and show how to run your code in README.