[MLDL for DS] Programming Assignment 02

MNIST is the most well-known dataset and also the simplest one that all students who start learning machine learning & deep learning must know. The dataset includes 60K training images and 10K testing images, wherein each image is a 28x28 gray image with digits from 0 to 9. Many researchers have tried to tweak this dataset such as moving MNIST or have introduced similar ones like English character dataset.
In this assignment, we will work with MNIST and EMNIST. The later one is a dataset consisting of 26 English characters A-Z. Each character has 6K training images & 1K testing images. We will learn how to use Pytorch to build complex neural networks and convolutional neural networks. It is not complicated to prototype models since most of the workloads are masked out inside Pytorch.

MNIST Demo 1993:
https://www.youtube.com/watch?v=FwFduRA_L6Q

1. Data description

	MNIST	EMNIST
Num Classes	10	26
Train|Test	60K | 10K	5K * 26 | 1K * 26
Content	Containing images of digits from 0 to 9	Containing images of English character from A - Z
Image size	28 x 28	28 x 28
Image quality	Gray, 1 channel	Gray, 1 channel

All datasets will be provided.

2. Problems

We will provide a template that has loading data functions, evaluation, and initialization. Your roles are only to focus on implementing NN and CNN models for MNIST & EMNIST.

Problem 1 (20 points): Neural Network Implementation for MNIST

Implement **a simple ANN (not CNN)** model that can predict digits accurately. Maximum number of hidden layers is three. Your model should get over 90% accuracy. If your model cannot achieve this level of accuracy, you get minus points in this problem.

- Try two different configurations (size of layers or number of hidden layers): **10 points**/each setting
  minus_points = round(2 * (20 - your_accuracy / 90 * 10))

Problem 2 (20 points): CNN Implementation for MNIST
Implement a **CNN** model that can predict digits accurately. Your model should get over 95% accuracy. If your model cannot achieve this level of accuracy, you get minus points in this problem.
minus_points = round(2 * (20 - your_accuracy / 95 * 20))

Problem 3 (20 points): MNIST CNN Ablation Study
Let's use the model implemented in problem 2 with different settings. For instance, you can change the kernel size. You must try at least 3 settings and report the results of each setting. After that, let's briefly explain reasons for your configurations. Do you have any interesting observations?

- Each setting tried & clear explanation: **5 points**
  - Vague description: minus **2 points**
- All settings get more than 95% accuracy: **5 points**

Problem 4 (10 points): CNN Implementation for EMNIST
As similar to problem 2, let's build a CNN model for EMNIST dataset. Your model should get over 70%. If your model cannot achieve this level of accuracy, you get minus points in this problem.
minus_points = round(2 * (20 - your_accuracy / 70 * 20))

Problem 5 (30 points): Transfer Learning
Transfer learning is an efficient technique that helps alleviate the lack of data in many applications. For instance, you train the model in a source task with an abundant amount of data then only retrain a few layers in a target task. Applying transfer learning to a specific problem sometimes requires much experience.
Let's try at least two transfer learning methods that load a trained CNN model on MNIST and retrain on EMNIST. Some well-known techniques are retraining all layers, retraining only the last layer, warm-up, or gradually unfreezing.

- Load model from MNIST -> EMNIST: **10 points**
- At least two settings: each setting gets **10 points**

- You need to report the results of each configuration and clearly explain your choice. Vague descriptions get minus from 2 points to 5 points in each setting.

P/S:

- **Discussion on algorithm and program logic is welcome but sharing code with classmates is strictly prohibited. Program code must be done by yourself.**
- **You will get full points if complete implementation and code can run. Depending on the level of completion, points of each problem can be varied. Let's submit to ETL no later than April 16th at midnight (23:59:59).**
- **For late submission, the penalty will follow the SNU's guideline (10% deduction each day, 0 after 5 days).**
- **Should you have any questions do not hesitate to contact TA (Alex):**
    - **Email: cuongbt91@snu.ac.kr Kakao Id: alexbui**