AI51601/CSE54501 Computer Vision Assignment 2 [100 points]

Deadline : until Dec. 19, 23:59

- If you have any questions, please post it in the blackboard/discussion/'assignment 2' board.
- Please use the PyTorch library when implementing CNN network. Other deep learning libraries (eg. Tensorflow, Keras) are not allowed.
- Please download the code here: https://drive.google.com/drive/folders/1azVvoCLd0FVezhe2qFu1aaDVuMVWV0_a?usp=sharing

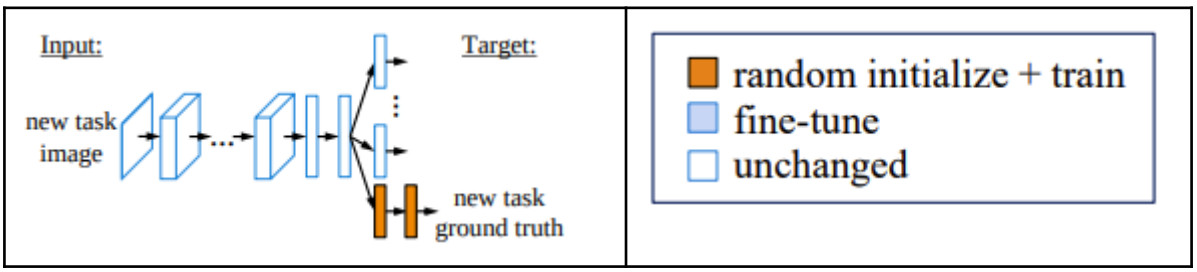
1. [30 points] Using the accompanying codes, please train the model defined in `model.py' on the MNIST dataset having 10 digits in an *incremental* manner using `normal_train.py': It will train the model for the first 3 epochs to classify the first 5 digits (ie. 0, 1, 2, 3, 4) using training data from the first 5 digits (ie. 0, 1, 2, 3, 4). Then, add additional outputs for remaining 5 digits (ie. 5, 6, 7, 8, 9) and train the model for another 3 epochs for new digits (ie. 5, 6, 7, 8, 9) using training data from the last 5 digits (ie. 5, 6, 7, 8, 9). The resultant model outputs a total 10 digits. In the CoLab, you may need to execute it as follows:

```
!pip install einops
!python normal_train.py
```

Please summarize the obtained results and explain them in the aspect of incremental (continual) learning.

[Tip: You may need to understand `cl_dataset.py' to know how `dataset.task_data' is initialized in `normal_train.py' for the incremental setting.]
[Tip: You may also need to understand the `adapt_last_layer' method in `model.py' to properly understand how `model.adapt_last_layer' does to extend the output of the model in `normal_train.py'.]

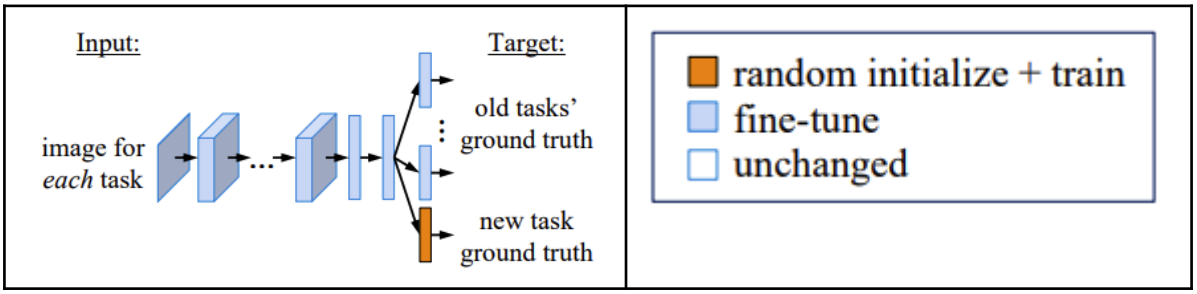
2. [15 points] Please implement `Prob2.py' for training the model as below figures ((1) adding a new layer for predicting new task labels, (2) freezing all networks except for the added new layer) and perform the incremental training samely as in Problem 1:



Please summarize results and analyze them.

3. [15 points] Please implement `Prob3.py' for training the model as below figures. ((1) train 3 epochs first using training data from the first 5 digits (ie. 0, 1, 2, 3, 4), (2) adding a new layer for predicting new task labels, (3) unfreezing all networks, and (4) train another 3 epochs using training data from all the digits (ie. 0, 1, 2, 3, 4, 5, 6, 7, 8, 9).

   The difference between Problem 1 and Problem 3 lies in the training data usage: In Problem 1, we do not use training data from the first 5 digits during the second 3 epoch training; while in Problem 3, we use training data from whole 10 digits during the second 3 epoch training. Please summarize and analyze results, by comparing them to results of Problems 1 and 2.



4. [15 points] Please implement `Prob4.py' samely as Problem 3. However, please implement it to use only (randomly sampled) 500 training samples for (0, 1, 2, 3, 4) digits; while using whole training samples for (5, 6, 7, 8, 9) in the second 3 epoch training: (1) train 3 epochs first using training data from the first 5 digits (ie. 0, 1, 2, 3, 4), (2) adding a new layer for predicting new task labels, (3) unfreezing all networks, and (4) train another 3 epochs using (randomly sampled) 500 training samples from the first 5 digits (ie. 0, 1, 2, 3, 4) combined with whole training samples from the second 5 digits (ie. 5, 6, 7, 8, 9).

   Please summarize and analyze how the final performance changes compared to Problem 1, 2 and 3.

5. [15 points] Please implement `Prob5.py' using any other training method for preventing catastrophic forgetting, which is different from the solutions in Problems 3 and 4. (The performance could be lower than those of Problems 3 and 4). Please summarize and analyze how the final performance changes compared to Problem 1, 2, 3 and 4.

6. [10 points] Please note and apply the below details.

   a. Send all files(.zip) via **Blackboard**.

   b. In **20231111_SeungryulBaek_cv_ass2.zip** file, you need to have four python files (*.py), a pre-trained network weight (.pth) and a report as follows:

<Example>

20231111_SeungryulBaek_cv_ass2.zip
```
├---Prob2.py
├---Prob2.pth
├---Prob3.py
├---Prob3.pth
├---Prob4.py
├---Prob4.pth
├---Prob5.py
├---Prob5.pth
└---Report.pdf
```

c. You need to write a report within 10 pages, properly summarizing your solutions or results for each problem.

d. Code should have some **comment** that increase readability(it is also grading points)

e. Your pre-trained weights in .pth need to be properly loaded.