Assignment 4
CSL7590: Deep Learning
AY 2024-25, Semester – II
Due on: 15-04-2025 11:59 PM

Maximum Marks: 50

General Instructions:

- Clearly mention the assumptions you have made, if any.
- Clearly report any resources you have used while attempting the assignment.
- Any submission received in another format or after the deadline will not be evaluated.
- Make sure to add references to the resources that you have used while attempting the assignment.
- Plagiarism of any kind will NOT be tolerated and will result in zero marks.
- Select your dataset correctly. If found otherwise, your assignment will not be evaluated.
- This is a group assignment, with a maximum group size of 3 students. **It is to be noted that the groups for each assignment should have unique team members, i.e., once you form a group with some students for any assignment, those students cannot be in your group for any future assignments in this course.**

Submission Guidelines:

- Prepare a Python code file for the task and name it as `RollNo1_RollNo2_RollNo3.py` (e.g. `M24CSE097_M24CSE098_M24CSE099.py` for a group with members M24CSE097, M24CSE098, and M24CSE099). There should only be one `.py` file in the submission. Do NOT prepare a separate `.py` file per subtask. The `.py` files must NOT be named like `<roll no>_task1(1).py`.
- Submit a single report depicting methods, results, and observations. There is NO need to add theory behind the concepts. Preparing a report is mandatory; failing it will lead to non-evaluation of the assignment.

- Name your report as `RollNo1_RollNo2_RollNo3.pdf`. Also, provide your Colab file link of your code in the report. Make sure that the Colab notebook access is shared to us/all.
- Do NOT make a zip file. Just upload both the code and the report directly on Google Classroom, i.e., the submission will contain `{RollNo1_RollNo2_RollNo3.py and RollNo1_RollNo2_RollNo3.pdf}`. Do NOT upload files in any other format.
- Do NOT download the `.ipynb` file, rename it as `.py`, and upload it. `.ipynb` files are not exactly in a readable form, so uploading it will only result in you receiving 0 marks for the same. You have an option to download a `.py` file in Google Colab directly.

- Do NOT copy-paste code or screenshots, etc., in the report. The report should look like a technical document, containing plots, tables, etc., whenever necessary.

Task: Implementing Adversarial Knowledge Distillation (AKD)

Objective:

In this assignment, you are required to implement **Data-Free Adversarial Knowledge Distillation (AKD)** using **PyTorch** to improve the performance of a student model by learning from a teacher model using adversarial training. The student model doesn't have any access to the training dataset so we only have to rely **on the teacher model** which is a larger model trained on the train set of the same dataset.

The core idea of this assignment is to use a generator to generate synthetic images which can be used to drive the student model's performance towards the teacher model's performance. You can consult **THIS PAPER** which introduced the idea of AKD.

(FOR FURTHER CLARIFICATIONS FEEL FREE TO REACH OUT TO THE TAs)

Dataset:

Use the **CIFAR-100** dataset only for testing purposes. It consists of 100 classes, with a standard split of 50,000 training and 10,000 testing images. **You will not be using the training data for the student model.**

Network Architecture:

A **pre-trained teacher model (ResNet-34, standard pytorch in-built implementation)** is provided which is already trained in the CIFAR-100 dataset. The weights of the model can be accessed from this **[LINK].** The accuracy of the teacher model on the train split of CIFAR-100 is **85.12%**.

To access the teacher model, you just need to declare the ResNet-34 in pytorch model for 100 classes and load the weights.

Students must design two student models:

1. One with approximately **10%(± 3%)** of the teacher's total parameters.
2. Another with approximately **20%(± 3%)** of the teacher's total parameters

Students are free to use any CNN model (or build their own as long as it satisfies parameter criteria)

Evaluation:

- Report for both student models (10% and 20%):
    - **Test Accuracy**
    - **Confusion Matrix**
    - **Total trainable and non-trainable parameters**
    - **Images generated by the generator network**

- Run experiments on **two data splits**:
    - 20% test data
    - 10% test data
    - **Reminder**: No training data is used directly to train the student — it is used only to train the teacher (already done) and for test evaluation.

Grading Rubrics:

- **Correct Implementation of Teacher & Student Model Training (10 points)**
- **Implementation of Knowledge Distillation architecture (20 points)**
- **Training & Evaluation for Multiple Train-Test Splits (15 points)**
- **Documentation and Clarity of Code (5 points)**
