AI504: Programming for AI (Fall 2024)

First Project: Image Classification

Start: Oct 22nd, 11:59:00 AM

Due : Oct 23rd, 11:59:00 AM

Project Instructions

In this project, you will perform a classification task on the Fashion-MNIST dataset, which contains 10 categories of fashion items, such as shirts, shoes, and pants. Your objective is to build a Convolutional Neural Network (CNN) that classifies these items based on the provided images.

We are providing a minimum baseline performance that you must surpass. We used Claude (free version) to write a simple 6-layer CNN without BatchNorm or Pooling layers. The resulting model consists of 4 convolutional layers and 2 fully connected layers. When tested in a Colab environment with **PyTorch version 2.4.1+cu121**, it achieved an accuracy of **92.15%** on the test set. Your goal is to design a model that surpasses this baseline.

Project Requirements

- **Dataset**
  - **Use the Fashion-MNIST** dataset, which can be downloaded and loaded as described in the following documentation: https://pytorch.org/vision/stable/generated/torchvision.datasets.FashionMNIST.html
  - **Important Note: Do not use the test set during training.** If you are found to have used the test set for training, it will result in an **automatic fail**.
- **Model Specifications**
  - You are free to design any CNN architecture, including pre-built models like ResNet or custom configurations.
  - **No restrictions** on model architecture, training process, or data preprocessing.
  - Your model must achieve an accuracy **greater than 92.15%** on the test set.

Submission Instructions

You must submit two files: one for the model's logits and one for the training script. These files must strictly follow the format below:

1. **Logits File (studentID.npy)**

- **Description**: In this project, logits refer to the raw, unnormalized output scores from your model for the **test set** before applying the softmax function. These scores represent the model's confidence for each class, where larger values indicate higher confidence.
- **Format:** A NumPy array with the shape **(10000, 10)**, corresponding to 10 classes for each of the 10,000 test images.
- **File Name:** Must be named as **studentID.npy** (e.g., 20241234.npy).
- **Important:**
  i. **Incorrect file names or formats will result in an automatic fail.**
  ii. The logits will be used to calculate accuracy using the provided test.py evaluation function. Make sure that your .npy file is compatible with this function. **Shuffling the test set is strictly prohibited, as it will invalidate your results**.

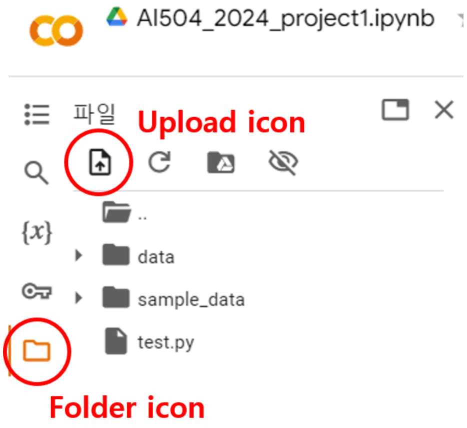
2. **Script File (studentID.py)**
   - **Functionality:** Your Python script should train the model and save the logits in a **single execution.**
   - **File Name**: The script must be named studentID.py (e.g., 20241234.py).
   - When we run python studentID.py, it should:
     i. Train your CNN model.
     ii. Save the logits as studentID.npy **in the same directory.**
   - **Important**: The .py file must save the .npy file directly without manual intervention.
3. **Submission Guidelines**
   - Upload the studentID.py and studentID.npy files to KLMS.
   - **Do not create folders or compress the files**.
   - **Grading will be based solely on the logits file**, but the script file is required to prevent cheating.
   - Failure to adhere to **any** of the above guidelines, including incorrect file names or formats, will result in an **automatic fail**.

Provided Resources

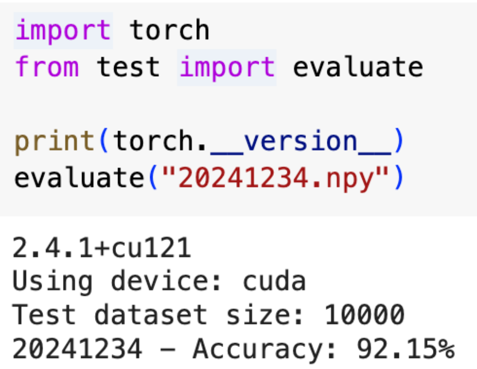
We have provided a **test.py** file to assist with your project. This file includes the **"evaluate"** function, which loads your logits file and calculates accuracy using the test set.

For students using **Colab**, follow these steps to use the **test.py** file:

1. **Upload the test.py File**:
   - In Colab, click on the folder icon on the left-hand side of the interface.
   - Click the "Upload" button and select the test.py file from your local machine.
   - This will upload the file to your Colab environment.



2. **Import the test.py File**:
   - After uploading, you can import the file by using the following command:
     // from test import evaluate
3. **Running Evaluation**:
   - Once you've generated your logits, you can call the evaluate function to check your model's accuracy // evaluate(“studentID.npy”)



```
import torch
from test import evaluate

print(torch.__version__)
evaluate("20241234.npy")
```

```
2.4.1+cu121
Using device: cuda
Test dataset size: 10000
20241234 - Accuracy: 92.15%
```

This process ensures you can seamlessly evaluate your model's performance in Colab using the provided test.py file.

Evaluation Criteria

- Your model will be evaluated using the test.py file. Accuracy will be calculated and **rounded to two decimal places** (e.g., 92.151 → fail, 92.155 → pass).
- Grading Criteria:
  - Pass: Accuracy greater than 92.15%.
  - Fail: Accuracy of 92.15% or lower, or if:
    - No submission is made.
    - The submission is late.
    - The test set is used during training.
    - **The submission does not follow the submission guidelines.**

Important Notes

- **Accuracy Metric:** Only the accuracy calculated using the **test.py** evaluation function will be considered for grading.
- **Project Support:** All project-related questions will be answered **only through KLMS** during the designated period: **October 22nd, 11:59:00 AM to 11:59:00 PM**.
    - **Emails will not be accepted.**
    - When posting questions on KLMS, please post them **publicly** so that others can also see them.
- **Submission Guidelines:** Failure to follow any part of the submission guidelines will result in an automatic Fail (grade F). No exceptions will be made due to the class size (approximately 500 students). Ensure your submission strictly follows the required format.