Machine Intelligence Lab(UE21CS352A)

Convolutional Neural Network

Dr. Uma D
Email: umaprabha@pes.edu
Teaching Assistants:
mohannitin28@gmail.com
pracheth.thamankar@gmail.com

A Convolutional Neural Network (CNN) is a specialized type of artificial neural network designed for processing grid-like data, particularly well-suited for tasks involving images and videos. CNNs have been pivotal in the field of computer vision, driving advancements in image classification, object detection, facial recognition, and more. At their core, CNNs employ convolutional layers that apply filters to localized regions of input data, enabling them to extract hierarchical features, starting from simple edges and textures and progressing to more complex shapes and objects. These networks also incorporate pooling layers to downsample feature maps and introduce non-linearity through activation functions like ReLU. Typically, fully connected layers are used for final predictions. CNNs can have varying depths with many layers, facilitating the learning of intricate patterns.

PyTorch is a top choice for CNNs due to its simplicity, dynamic computation graph for flexible model design, strong community support, and GPU acceleration, making it ideal for efficient and effective development of CNN-based applications.

Task:

Complete code for functions whose skeleton has been provided.

You are provided with the following files:

1. CNN.py
2. Test.py

CNN.py

This file contains the following functions:

Function name	Input	Output
init	1. self	-
forward	1. x: ndarray/tensor (img)	prob_list: sequence of 3 probabilities
train	1. model: CNN Object	train_accuracy: float

Test.py

1. This will help you check your code.
2. Rename CNN.py file to CAMPUS_SECTION_SRN_Lab3.py
3. Run the command `python3 Test.py --ID CAMPUS_SECTION_SRN_Lab3`

The sample dataset used is a subset of the CIFAR-10 dataset available in PyTorch. The subset contains images belonging to three classes (1,8,9) that have been reindexed to (0,1,2). So consider that the dataset has three classes (0,1,2). You can define a maximum of 8 Layers in init function and stack them appropriately in forward.

Note: The model must have training accuracy (>=60%), and testing accuracy(>=70%).

Important Points

1. **Do not make changes to the function definitions** that are provided to you. Use the skeleton as it has been given.
2. **Do not make changes to the test file provided to you**. Run as is.
3. Do not hardcode values. The functions must be designed to be independent of the schema of the dataset.
4. You may write additional helper functions.
5. You **can use built-in modules**.
6. You **must use PyTorch**

Submission Guidelines

You are to submit two files:

1. The python solution: CAMPUS_SECTION_SRN_Lab3.py
2. Screenshot of Test cases (Single screenshot of all testcases in terminal): CAMPUS_SECTION_SRN_Lab3.png (or jpg)

The google form link for submission will be provided

- Remove all print statements.
- Failing some hidden cases will lead to partial marks.
- Test cases provided to you are for reference only, hidden test cases will be similar