CS776: Deep Learning for Computer Vision
Assignment 1
Due Date: February 2, 2022 (11:59 PM)

General Instructions:

- This is an individual assignment.
- Only electronic submissions will be accepted. your solution has to be submitted via the mookit (Hello IITK) platform.
- Late submissions will not be accepted.
- Any sort of plagiarism will be penalised. If you are referring to any materials online or books, please cite them accordingly otherwise it will be considered as plagiarism.
- You should not use built-in functions from Numpy or PyTorch or other image processing libraries for image augmentation/transformation. You need to write your own functions for this assignment. For more details read the instructions in the question.
- Please ensure that the report is written as per the instructions given in the assignment.
- You are free to use systems from KD Lab or Google Colab or Kaggle to train the models.
- Instructions for KD Lab machine login:
  1. Connect vpn: mention server → gateway.iitk.ac.in and port → 443, cc-userid, cc-userpasswd
  2. from a new terminal: ssh cse-userid@172.27.19.x (1-100)
  3. enter cse-passwd
  4. to keep the remote terminal running even after closing the local terminal use screen or tmux
  5. to init a screen: use screen -t [some name: alphanumeric]
  6. to get back to the last screen: use screen -x (if you have initiated multiple screen then screen -x will return some ids, then to go back the corresponding screen use: screen -x [id])
  7. For installing packages on the KD systems from internet, you need to disable the firewall. Run **authenticator.py**, a python3 script provided with the assignment to disable the firewall after logging into any of the KD system. You can keep this terminal open. Now, open another terminal to do the package downloading required for the assignment.

Question (100 Marks)

In this assignment, your goal is to implement and train a multi-layer perceptron (MLP) on the CIFAR-10 dataset with data augmentation (Single hidden layer with 64 neurons MLP model).

1. Download the CIFAR-10 dataset (`https://www.cs.toronto.edu/~kriz/cifar.html`). Hint: Use pickle library to load the dataset. [5 marks]

2. Implement the image transformation methods mentioned below: [5x4=20 marks]

   (a) Random Rotation in the range $[-180°, 180°]$

   (b) Random cutout (randomly erase a block of pixels from the image with the width and height of the block in the range 0 to 16 pixels. The erased part (cutout) should be filled with a single value)

   (c) Random Crop (Add a padding of 2 pixels on all sides and randomly select a block of 32x32 pixels from the padded image)

   (d) Contrast & Horizontal flipping. (First, change the contrast of the image with a factor of α randomly selected from the range $(0.5, 2.0)$ and then flip the image horizontally with a probability of 0.5)

   Instructions for changing contrast of an image:
   Given an image x with pixel values in range [0, 255], you can change its contrast by a factor of α using the following steps:

   i. Change all the pixel values with formula $x'(i, j, c) = \alpha \cdot (x(i, j, c) - 128) + 128$. Here, $x(i, j, c)$ refers to the c-th channel value of the (i, j)-th pixel.

   ii. Clip the pixel values in x' so that the final values are in the range [0, 255]

   Note that $\alpha < 1$ will decrease the contrast of the image while $\alpha > 1$ will increase the contrast.

   For each transformation, define a python function that takes an input image and returns the transformed image. Use of any built-in functions (such as pillow, sklearn, PIL, CV2, etc.) is prohibited for this part. For all the padding operations as well as the gaps created due to the image transformation operations, fill with zero values. Demonstrate each transformation method by applying the transform functions on at least one example image.

3. Create the augmented training set using the transformation functions implemented in the previous part. Randomly select one of the four transformations for each image in the training set and apply it to that image. Combine the transformed images with original training set to get the augmented training set. Note that the number of examples for the augmented training set will be twice that of the unaugmented training set. [10 marks]

4. Use the **feature_extractor.py** file provided with the assignment on the original (unaugmented) CIFAR-10 dataset and on the augmented dataset to get 1-dimensional input vectors. You can ignore the implementation of **feature_extractor.py** and use it directly. [10 Marks]

   Instructions for using feature-extractor.py:

   - Refer to this page : https://pytorch.org/get-started/locally/ for installing the required dependencies of PyTorch. However, if you are using Google Colab or Kaggle for running your code then you do not need to install PyTorch there, as these environments supports PyTorch.

- The feature_extractor.py accepts images of size $(3\times224\times224)$ [Channel × Height × Width]. Use image processing libraries like PIL, CV2 to resize the CIFAR images from $(3\times32\times32)$ to $(3\times224\times224)$.
- Pass the resized images to **feature_extraction** function of BBResNet18 class to generate feature vectors
- **feature_extraction:** function expects each image is a **numpy.ndarray** of dtype: **numpy.float32** and shape: **[None, 3, 224, 224]**, where: **None** represents a variable size. It returns a **numpy.ndarray** of dtype: **numpy.float32** and shape: **[None, 512]**.

5. Implement a multi-layer perceptron (MLP) for classification of CIFAR-10 images. Use only a single hidden layer with 64 neurons and ReLu activation function. The input to this MLP will be the 1-dimensional vectors generated in the previous step. [10 Marks]

6. Implement the back-propagation algorithm and use it to train the MLP model on: [20 marks]

   (a) original training set

   (b) augmented training set

   You are not allowed to use built in functions that performs back propagation directly. You should write your own back propagation algorithm.

7. Evaluate the performance of the following trained MLP models on the original (unaugmented) test set: [10 marks]

   (a) MLP model, trained on original training set

   (b) MLP model, trained on augmented training set

8. Submit a report clearly explaining how you have built the models, the architecture of the models, learning rate, epochs used for training, evaluation metrics and the instructions for running the models. Derive the gradients and update expressions needed to implement back-propagation, Compare the performance of the two models on the original (unaugmented) test set and justify the observed behaviour. [15 marks]

Deliverables:

1. The solution for the assignment should be submitted as a zip file. The file should be named as StudentNameRollNumber.zip.

2. The submission should contain the following:

   - A python file implementing image transformation functions and creating augmented dataset using the transformations
   - MLP model implementation
   - The report (as pdf).