CS 564: Foundations of Machine Learning
Assignment 4

Deadline: 30 November 2022

- Markings will be based on the correctness and soundness of the outputs.
- Marks will be deducted in case of plagiarism.
- Proper indentation and appropriate comments (if necessary) are mandatory.
- Use of frameworks like PyTorch, TensorFlow etc. is allowed.
- All benchmarks (accuracy etc.), answers to questions and supporting examples should be added in a separate file with the name 'report'.
- All code needs to be submitted in '.py' format. Even if you code it in '.IPYNB' format, download it in '.py' format and then submit
- You should zip all the required files and name the zip file as:
    - <roll_no>_assignment_<#>.zip, e.g., 1501cs11_assignment_01.zip.
- Upload your assignment (the zip file) in the following link:
    - https://www.dropbox.com/request/57BCyzMnwjUlS4OMBBah

Dataset:
For this assignment, we'll be using the CIFAR-10 dataset. The download link is: https://www.dropbox.com/s/q2dms7ebgkacj5c/cifar-10-python.tar.gz?dl=0

Details of the dataset can be found at https://www.cs.toronto.edu/~kriz/cifar.html . Use the steps in the details link to unpack the data. Alternatively, you can download the dataset via:

- PyTorch:
    - from torchvision.datasets import CIFAR10
- TensorFlow:
    - from tensorflow.keras.datasets import cifar10

The dataset consists of 60000 images (train-test split of 50000-10000) of 3×32×32 size. The task is to label the image as one of the 10 output classes.

Problem Statement: Design and implement a Feed Forward Neural Network (FFNN) and a Recurrent Neural Network (RNN) for the task of image classification on the CIFAR-10 dataset.

Implementation Details:

1. *Model input:* 3×32×32 size image, *model output:* image label/class (total 10).
2. *Validation set:* The dataset consists of 50000 training images, which are to be split in 9:1 ratio for 45000 training images and 5000 validation images.

3. *Loss and optimizer:* use NLL (negative log likelihood) loss and stochastic gradient descent optimizer.
4. *Hyperparameters:*
    i. Use 1024 as the hidden layer size for FFNN, over each next layer and halve the hidden layer size until you reach the hidden size of 32 (i.e., layer1=input_size x 1024, layer2=1024 x 512, … , 32 x output_size). Overall, the FFNN will have 7 layers (layer1=input_size x 1024, layer2=1024 x 512, layer3=512 x 256, layer4=256 x 128, layer5=128 x 64, layer6=64 x 32, layer7= 32 x output_size)
    ii. Use 1024 as hidden layer size for RNN, over a total of 3 hidden layers.
    b. The final output size would be 10 (same as the number of classes) for both models.
    c. The final output must pass through a softmax layer.
    d. The batch size would depend on your memory limitations.
    e. Train the model for 50 epochs.
    f. Assume other hyperparameters as per your intuition.
5. **Evaluation:** Report the following in your submission
    a. Loss and accuracy for training phase (on validation set) of FFNN and RNN
    b. Loss and accuracy for testing phase (on test set) of FFNN and RNN
    c. Plot the loss and accuracy for both the cases above.
    d. Write reasoning for why an RNN works better than FFNN in general.

Documents to Submit:

1. Model code
2. Submit Test Set Predictions
3. Write a report (doc or PDF format) on how you are solving the problems as well as all the results, including model architecture.

NOTE: We are working with image datasets which are a matrix of numbers (in this case a 3×32×32 matrix). Since the input is already in numbers, there is no need to add a feature vector for the intent and purpose of this assignment. Directly feed the input to the first layer of both the models (1024 in both cases)

For any queries regarding this assignment, contact:

- Abhisek Tiwari (abhisektiwari2014@gmail.com),
- Ratnesh Kumar Joshi (ratneshkr.joshi@gmail.com), and
- Ramakrishna Appicharla (ramakrishnaappicharla@gmail.com