Scalable Data Mining (Autumn 2021)

Assignment 2: Pytorch (Full Marks: 100)

Question 1: (100 marks)

Task: The aim of this assignment is to train and test Convolutional Neural Networks for image classification on CIFAR10 dataset using PyTorch Module and to acquaint yourself with wandb which is a tool for monitoring large experiments.

Data:

- train_images: Consist of 50000 images of 32 x 32 RGB images.
- train_labels: Consist of 50000 labels from 10 classes for the images in train_images. The labels are described below.
- test_images: Consist of 10000 images of 32 x 32 RGB images.
- test_labels: Consist of 10000 labels from 10 classes for the images in test_images.

Labels: Each training and test image is classified into **ANY ONE** of the following labels:

0 - Airplane	1 - Automobile
2 - Bird	3 - Cat
4 - Deer	5 - Dog
6 - Frog	7 - Horse
8 - Ship	9 - Truck

Implementation Details:

Supporting files are available here: https://github.com/SoumiDas/CS60021_A2021

1. **Data Load:** Use the file **main.py** given in the above link to load the data and carry on further experiments.

2. **Model**: Use the pretrained Resnet18 model to train your Convolutional Neural Network in the following ways.
   a) **Train all the layers** **b) Freeze the other layers and finetune only the last layer**.

You may get differences in accuracy in the above two methods. Describe what is the reason for the same.

3. **Training Module**: Implement a mini-batch SGD using **main.py** to train the CNN in both the ways described above. Use the following configurations while training:

- ● Use SGD optimizer with learning rate = 0.001, momentum = 0.9 and cross-entropy as the loss function.
- ● Use Adam optimizer with learning rate = 0.01 and cross-entropy as the loss function.

You can use early stopping too if loss converges beforehand.

For each of the two configurations above under each way (described under Model), retain/save the best model (yielding best test set accuracy while testing at each epoch).
Use the best saved models to report the final test set accuracies for all the four combinations.

Submission Details:

You should submit the following in zipped format (Rollnumber_AssignmentNo.zip):

- **Report** as asked below (Report.pdf) using wandb (details below) with all the contents as mentioned under the '**Submission Details**'. Analyse the observations and explain them. **(60 marks)**
- **Python codes :** main.py - you can add functions as per requirements. **(40 marks)**

Your report should contain the following details:

1. For both the configurations (under Training Module) under each method (under Model), plot a visualization diagram of Confusion Matrix for all the 10 classes in the test set and indicate which class got classified the best and the worst among all.

Note: You will basically be having 4 confusion matrices in total (2 configurations for each method under **Model**).

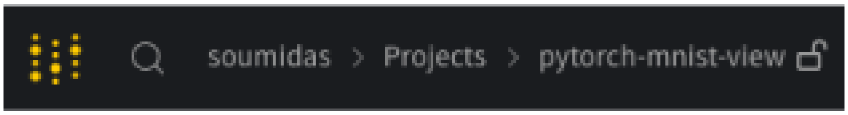
2. **Wandb:** Log all the results (training loss vs epochs, test loss vs epochs, test set accuracy vs epochs, sample prediction samples) in the wandb dashboard and create a report of the same. Also, mention the observation behind difference in accuracies (if any) after training in the two ways described under **Model**.

Additional information

A tutorial of training CNN on MNIST and using Wandb to log results can be found in this link: https://colab.research.google.com/drive/1XBC88XMqMFpl7AenpHXPnFhhAQN8pYiQ?usp=sharing

a. A sample report generated from wandb artifacts can be found in https://www.dropbox.com/s/nofyinh8k5ze30a/MNIST%20using%20CNN%20in%20Pytorch%20_%20pytorch-mnist-view%20%E2%80%93%20Weights%20%26%20Biases.pdf?dl=0 .
b. The link for the report can be also viewed in https://wandb.ai/soumidas/pytorch-mnist-view/reports/MNIST-using-CNN-in-Pytorch--Vmlldzo2Njg0Mjg .
c. Some guide on creating nice reports can be found in this link https://docs.wandb.ai/guides/reports

Please make sure to make the project “public” under which you’re saving the artifacts. A sample image of what you can see on your wandb profile when you view your project in public mode:



Project name for this case is **pytorch-mnist-view** . Please note the lock sign beside it which indicates it’s public. Please make the wandb report organised and explain your observations.

==