**Homework 1 Student Name: Chinmay Samak**

AuE 8930: Computing and Simulation for Autonomy

Instructor: Prof. Bing Li, Clemson University, Department of Automotive Engineering

\* Refer to [Syllabus](https://tinyurl.com/computing-autonomy) for homework (late) submission, grading and plagiarism policies;

\* Submission due Mon. 10/02/2023 11:59 pm via Canvas, include:

* This document (with answers), and with your program results/visualization;
* A .zip file of (modified) source code and data if any, which the TA might run.

**Question 1**

Training a Pytorch deep learning model on Palmetto cluster (60 points)

(Recommended to use Jupyter Notebook in Palmetto [OpenOnDemand](https://openod.palmetto.clemson.edu/) for edit/debug/run)

**Palmetto Cluster and Setup**

* Login into your Palmetto account & request a node with required specifications by specifying a hardware resource configuration, making sure to include GPU. (For below all questions, make sure to use same configuration).
* Transfer the sample code into your account using Globus (if using Terminal) or JupterHub.

**Create a Conda virtual environment in the terminal**

*module add anaconda3/2022.05-gcc/9.5.0*

A conda virtual environment allows you to run/install a version of Python and package as needed within it.

This environment, once created/modified is saved and can be accessed later through the code:

*conda create -n NAME\_OF\_ENV python=3.6 # (Create Environment)*

*source activate NAME\_OF\_ENV # (Activate Environment)*

*source deactivate NAME\_OF\_ENV # (Deactivate Environment)*

**Install necessary packages in the terminal**

Add cuda and cudnn module:

*module add cuda/11.1.1-gcc/9.5.0*

*module add cudnn/8.0.5.39-11.1-gcc/9.5.0-cu11\_1*

Install Pytorch and Torchvision libraries using conda ([reference](https://pytorch.org/get-started/locally/))

*conda install pytorch torchvision torchaudio cudatoolkit=11.1 -c pytorch-lts -c nvidia*

**Generate Kernel for JupyterHub**

(Attention: if you install those modules under a certain conda environment)

You may encounter this error when running the base.ipynb in Jupyter Hub:

"no module named torch"

It means your Jupyter notebook is running in the default python environment, but your torch module is installed in your Conda virtual environment. You will need to run Jupyter notebook in your virtual env.

Here is a tutorial: <https://janakiev.com/blog/jupyter-virtual-envs/>

**Training deep learning model for Image Classification**

Sample code is in Canvas/Files can be downloaded from: [Homework\_1\_sample\_code.zip](https://clemson-my.sharepoint.com/:u:/g/personal/bli4_clemson_edu/EYxlIdnUZU5KrcKe5qJSvYoB8LWCaijiVT1dV4pc2Y3j9g?e=EulwYv)

which includes: base.ipynb, common.py and models.py. The base.ipynb allows you to use your web browser as the GUI to run/edit/debug.

You also need to make ‘data’ and ‘models’ folder before running the ‘base.ipynb’. The directory structure should look like:

Table

Description automatically generated with low confidence

There are multiple steps in the sample code files:

* Load the training and test datasets from torchvision ([reference](https://pytorch.org/vision/stable/datasets.html))  
  Training Data can be obtained from various online sources, self-procured or can even be imported from a library like Pytorch.
* Define a Convolutional Neural Network ([reference](https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148))
* Define a loss function ([reference](https://blog.algorithmia.com/introduction-to-loss-functions/))
* Train the network on the training data with different number of Epochs ([reference](https://towardsdatascience.com/epoch-vs-iterations-vs-batch-size-4dfb9c7ce9c9)).

(1) Show screenshots of successful installation and procedure of the setup. (15 points)

A screenshot of a computer

Description automatically generated

Fig. 1. MobaXterm SSH Setup for Connecting to Palmetto Cluster

A screenshot of a computer

Description automatically generated

Fig. 2. Login Authentication

A screenshot of a computer screen

Description automatically generated

Fig. 3. Successful Login

A screen shot of a computer program

Description automatically generated

Fig. 4. Clone Course Git Repository and Verify Path to and Existence of the Cloned Repository

A screenshot of a computer

Description automatically generated

Fig. 5. Create Directory for HW1 and Make it Working Directory

A screenshot of a computer

Description automatically generated

Fig. 6. Copy HW1 Baseline Files to Palmetto Cluster

A screenshot of a computer

Description automatically generated

Fig. 7. Submit Interactive Job on Palmetto Cluster with name “aue8930”, 16 CPU Cores, 1 GPU (V100 Model), 32 GB RAM and 100 Gb Interconnect with 8 Hours of Wall Time

A screenshot of a computer

Description automatically generated

Fig. 8. Add Anaconda Module

A screenshot of a computer screen

Description automatically generated

Fig. 9. Create Conda Environment

A screenshot of a computer screen

Description automatically generated

Fig. 10. Add CUDA and cuDNN Modules

A screenshot of a computer

Description automatically generated

Fig. 11. Activate Conda Environment and Install Pytorch

A screenshot of a computer

Description automatically generated

Fig. 12. Continue to Palmetto OnDemand for Launching Jupyter Notebooks Easily with Lag-Free Experience (PFA the detailed job configuration in Appendix 1 at the end of this document)

A screenshot of a computer

Description automatically generated

Fig. 13. Create Jupyter Notebook Session on Palmetto OnDemand (wait for the job to start “Running” and then hit “Connect to Jupyter”)

A screenshot of a computer

Description automatically generated

Fig. 14. Open the Baseline HW1 Code

(2) Run the existing sample code “base.ipynb” (5 points)

During the training, what’s your GPU usage percentage? (You can open another terminal and use “nvidia-smi –l” to monitor the usage info of GPU and GPU memory.)

A screenshot of a computer

Description automatically generated

Fig. 15. Install “ipykernel”, Add “aue8930” Conda Environment and Run Jupyter Notebook in this Environment

A screenshot of a computer

Description automatically generated

Fig. 16. Start New Terminal Session from Launcher

A screenshot of a computer

Description automatically generated

Fig. 17. Check the GPU Usage [One GPU] as the Network Trains

**The GPU Usage Percentage was: 22% (GPU RAM: 1927 ÷ 32768 × 100 = 5.88%)**

(3) Modify the code for better performance (change the batch size) (10 points)

During the training, what’s your GPU usage percentage?

A screenshot of a computer

Description automatically generated

Fig. 18. Duplicate “base.ipynb” and Modify Batch Size to 256 for Faster Training

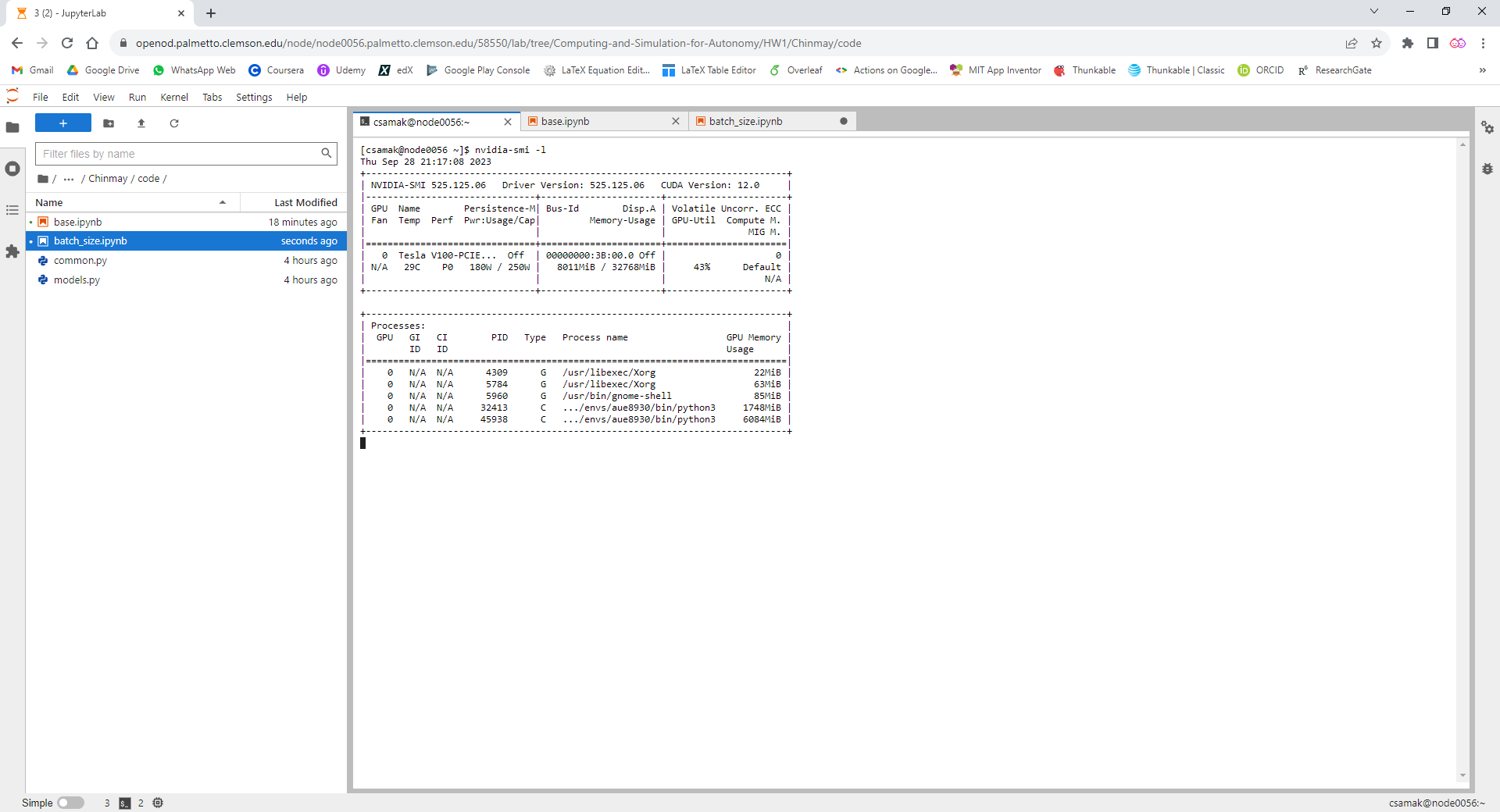


Fig. 19. Check the GPU Usage [Batch Size = 256] as the Network Trains

**The GPU Usage Percentage was: 43% (GPU RAM: 8011 ÷ 32768 × 100 = 24.44%)**

(4) Modify the code for better performance (use two GPUs) (10 points)

During the training, what’s your GPU usage percentage? (TIPS: [reference API](https://pytorch.org/tutorials/beginner/former_torchies/parallelism_tutorial.html))

A screenshot of a computer

Description automatically generated

Fig. 20. Start New Palmetto OnDemand Session with 2 GPUs

A screenshot of a computer

Description automatically generated

Fig. 21. Duplicate base.ipynb and Execute Training with 2 GPU Cores

A screenshot of a computer

Description automatically generated

Fig. 22. Use DataParallel to Utilize 2 GPU Cores Simultaneously

A screenshot of a computer

Description automatically generated

Fig. 23. Check the GPU Usage [Two GPUs] as the Network Trains

**The GPU 1 Usage Percentage was: 39% (GPU RAM: 3546 ÷ 32768 × 100 = 10.82%)**

**The GPU 2 Usage Percentage was: 15% (GPU RAM: 1606 ÷ 32768 × 100 = 4.9%)**

(5) Plot the accuracy against the number of training Epochs on a Graph. (10 points)

(TIPS: you need to import matplotlib, modify the code of “for epoch in range (EPOCHS):” by saving the “epoch” and “train\_acc”, and plot its relationship in the end)

A screenshot of a computer

Description automatically generated

Fig. 24. Modify Code to Plot Training Accuracy Against Epochs

(6) Could you improve on the network model, train it for better accuracy? (optional, 5 points)

(This question is optional. Extra 5 points until reach the cap of 100)

A screenshot of a computer

Description automatically generated

Fig. 25. Hyperparameter Tuning

The best accuracy was obtained with the following set of hyperparameters:

* Epochs = 100
* Batch Size = 256
* Optimizer = Adam
* Learning Rate = 0.001

Furthermore, following hyperparameters resulted in a more stable training in general and a much faster convergence:

* Epochs = 100
* Batch Size = 256
* Optimizer = Adam
* Learning Rate = 0.001
* Scheduler = LambdaLR (Lambda = 0.65#Epoch)

A more thorough hyperparameter tuning could be done systematically using full-factorial or Latin hypercube sampling methods so as to cover a broader hyperparameter set with many permutations (which is beyond the scope of this assignment).

(7) Perform a model inference for a certain image, which you can choose from anywhere. The image shall include the object which belongs to the category of the training dataset. (10 points)

(TIPS: if you are using CIFAR10 datasets, its categories are shown in this [reference](https://www.cs.toronto.edu/~kriz/cifar.html))

A screenshot of a computer

Description automatically generated

Fig. 26. Perform Model Inference for Test Images (from CIFAR-10 dataset)

**Question 2**

Write a 2~3 pages survey report on a particular High-Performance-Computing application related to engineering/vehicles (40 points). The grading of this question will be based on the contents which the survey covers:

- What is the problem to be solved (5 points);

- The importance of the problem to be solved (5 points);

- The challenges of solving this problem (10 points);

- Existing solutions of solving this problem (15 points);

- Other grading factors (such as novelty, organization, etc.) (5);

\* You are encouraged to include any drawing/table in the report;

\* Attention: use like [1] to cite a content you referred to, with reference list in the end. You should never literally copy contents from other places;

TIPS: you should survey and read multiple academic papers. Then, summarize for the above.