EE 554 Fall 2024 Lab 3

Total Points: 25

Student ID: _____ Name: _____

Assigned: 10/18/2024

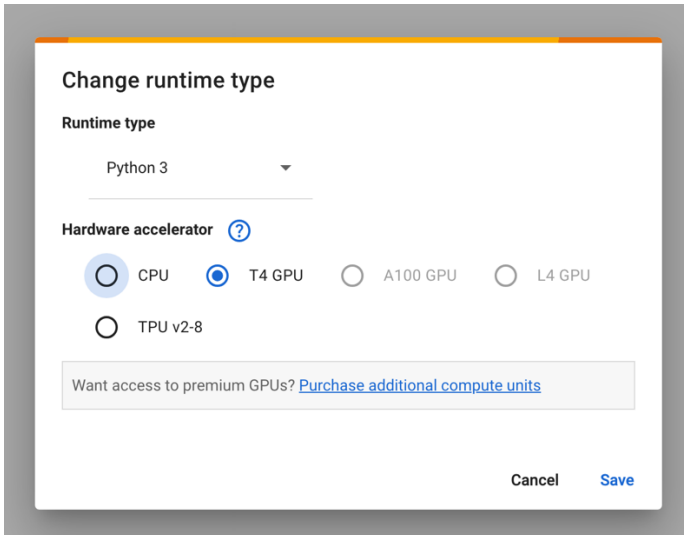
Due: 11/01/2024 No late submission accepted for this lab. This lab should be done individually.

1. Objective

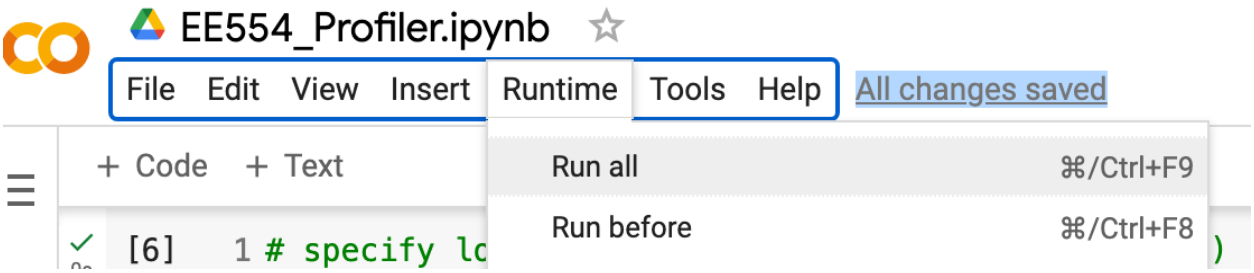
   PyTorch is an open-source deep learning framework widely used for developing and training machine learning models, particularly in the fields of computer vision and natural language processing. It provides a flexible and dynamic computational graph that allows for easier debugging and experimentation. PyTorch's main features include automatic differentiation, GPU acceleration, and an intuitive Python interface, making it accessible to both researchers and developers.

   The purpose of this assignment is to help you get acquainted with PyTorch model inferencing and Profiling to deploy and observe the performance of your PyTorch model on supported HW platforms.

2. Go through **Mini_lab for ML** in Brightspace if you do not have any experience with ML.
3. All the example for profiling is written in https://colab.research.google.com/drive/1dbglqRAVFTHG3-MvAKQQxhq3FAcqxThl?usp=sharing
4. Open the above file and change runtime type to T4 GPU in the Runtime tab on the top left.



5. Select Run all under Runtime tab.

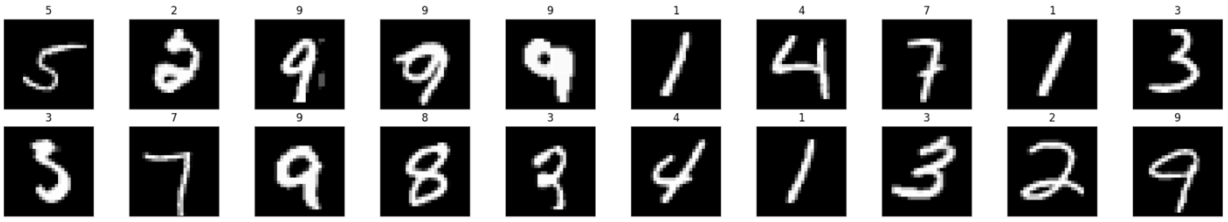
6. 

7. Deliveries:

Each student is expected to submit a report with following contents:

Notice: Google Colab is an easy way to run the PyTorch code. If you want to run the source in your own environment, it is fine. Make sure your code could be run in Google Colab with T4 GPU and share your link in the report. This is the only environment we will test your source code. Unsuccessful run will result in 80% deduction on the score.

a. Write Resnet50 model in PyTorch with Google Colab. Share the link of your code in your report.
b. Clearly define and explain what your input data is like. For example, in the example we provided:



c. Show your training, validation, and testing results. Accuracy and loss.
d. Generate two inference "trace.json" files and plot it in the https://ui.perfetto.dev/. Include the two table and two screenshot in the report. One trace should run 100% on CPU and One trace should have some operator in GPU.
e. Virtualized the profiler result in Google Colab with tensorboard for inference on a CPU-GPU platform. Took a screenshot of the trace with the relationship between an operator and its launched kernels.
f. Explain what Operator use the most of GPU time and why.
g. Optimized your inference run as indicated by Performance Recommendation. Report how you do it and what is the change in terms of inference time and memory usage. Include a screenshot of the Overview.