**EIE4122 Deep Learning and Deep Neural Networks**

**Lab 1: CNNs for Handwritten Digit Classification**

**A. Objectives and Outcomes**

After finishing this lab, you should be able to perform the following:

* Use Tensorflow and Keras to develop convolutional neural networks for handwritten digit classification.
* Use PyTorch to develop convolutional neural networks for handwritten digit classification.

**B. Assessment Criteria**

* Ability to build pattern classification systems based on Keras, TensorFlow, and PyTorch.
* Ability to produce correct results.
* Ability to change the programs and obtain meaningful results.
* Ability to explain the observations and results.
* Clarity of the report.

**C. Submission**

* Copy and paste the graphs and images that you obtain from Google Colab or other Python IDE to a word file and discuss your observations and results.
* **Convert your file to PDF**.
* Submit your report to Blackboard before the deadline specified in Blackboard.

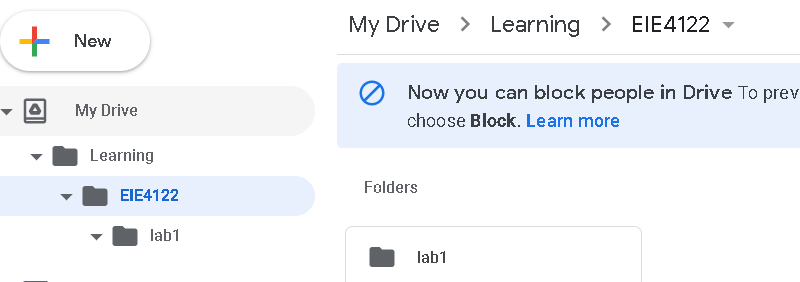
**D. Procedures**

***D.1 Prepare Colab Environment***

1. Colab runs on browsers. Google Colab is a free cloud service with GPU support. You may use Colab to develop deep learning applications based on popular libraries such as Keras, TensorFlow, PyTorch, and OpenCV. You need a Google account to use Colab. If you do not have one, visit https://support.google.com/mail/answer/56256?hl=en.
2. Display the Google Drive (https://drive.google.com/drive/) page in your browser. Create the following directory structure in your Google Drive:

My Drive/Learning/EIE4122/lab1/

After creating the folders, you should see something like this:

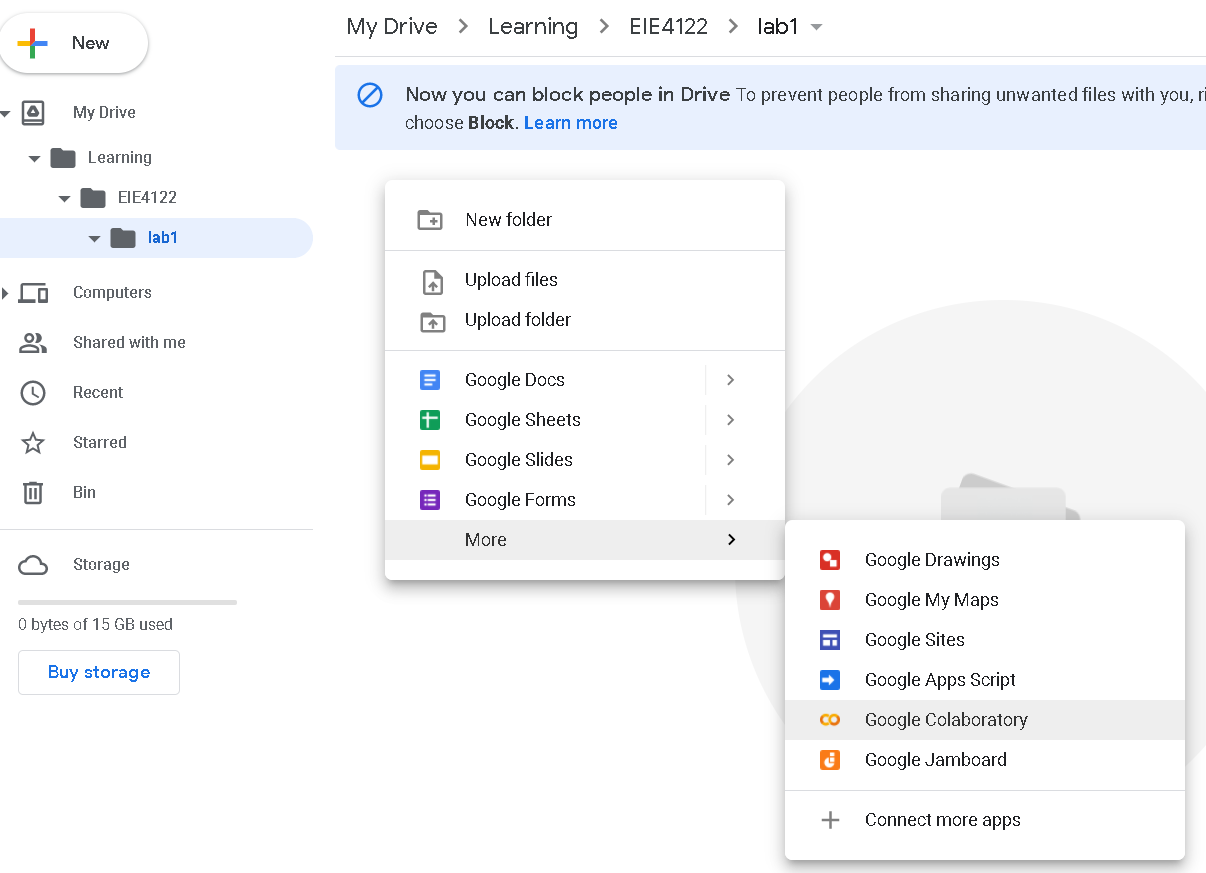


Directory structure in Google Drive

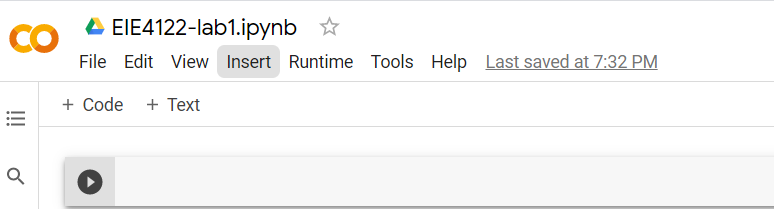
1. Download the files “mnist\_cnn\_keras.py” and “mnist\_cnn\_pytorch.py” from <http://bioinfo.eie.polyu.edu.hk/download/EIE4122/lab1> and put them into

“My Drive/Learning/EIE4122/lab1/” in your Google Drive.

1. In your Google Drive page, go to “My Drive/Learning/EIE4122/lab1/”. Then, right click on the empty folder, select “More” and then “Google Colaboratory” to create a Colab ipython file as shown below. If your browser does not have “Google Colaboratory” installed, click “connect more apps” and search “Colaboratory” to install it.



Change the IPython Notebook file to “EIE4122-lab1.ipynb” as follows:

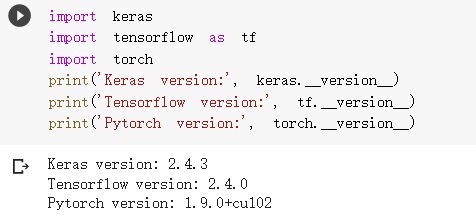


1. Install probable version of TensorFlow and Keras.



Note that if your session is time out and you reconnect to Colab, you need to perform this step to reinstall Tensorflow 2.4.0.

1. Check Keras, Tensorflow and PyTorch version. Click “+ Code” to create a new command edit box and execute the following:



Note that you need to click the Run button  to execute this command.

1. Click “+ Code” to create a new command edit box. Then, mount your Google Drive to the IPython Notebook as follows:

A screenshot of a cell phone

Description automatically generated

Click the link and follow the instruction. Put the key in the edit box and press the “Return” key. You should be able to see the following after mounting.

A screenshot of a cell phone

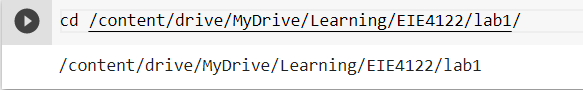
Description automatically generated

1. Configure Colab to use GPU by clicking Edit 🡪 Notebook settings. Select “GPU” in the pop-up window:

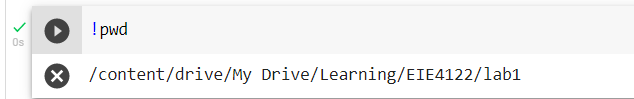
Graphical user interface, text, application, email

Description automatically generated

1. Change your current working folder to “My Drive/Learning/EIE4122/lab1/”:



1. Check your current folder:



***D.2 DNN for Handwritten Digit Recognition***

***Capture experimental results and add the results to your report.***

1. Execute the script “mnist\_cnn\_keras.py” to train a CNN implemented by TensorFlow (keras):



Read the code in “mnist\_cnn\_keras.py” carefully to understand how to implement and train a CNN using TensorFlow and Keras. Note that you can double click the file “mnist\_cnn\_keras.py” under the folder “Learning/EIE4122/lab1” on the left panel of Colab. Then, you may see an edit window that allows you to edit the python file.

The script also saves the CNN in .h5 format in the folder “models/”. Change the number of layers or the number of nodes to investigate how these parameters affect performance. Also, change the activation function to see if you can get better performance. Noted that you can change “model\_path” to store the new model.

1. Execute the script and use TensorBoard to visualize training details:

Text

Description automatically generated with low confidence

Note that TensorBoard is a tool allowing researchers to measure the performance and visualize the network’s activities during training. It allows tracking experiment metrics such as loss and accuracy, visualizing the model graph, projecting embeddings to a lower dimensional space, and much more.

1. Execute the script to test a CNN implemented by TensorFlow and Keras:



The script loads the model from ‘models/mnist\_cnn.h5’ that you have trained in the previous step. Note that you should specify ‘model\_path’ to load the model that you want to test if you have different models.

1. Install TensorBoard for pyTorch:



1. Execute the script to train a CNN implemented by PyTorch:



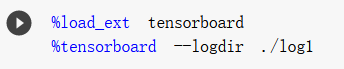
The script also saves the CNN in PyTorch format in the folder “models/”. Go through carefully the code in “mnist\_cnn\_pytorch.py” and understand how to implement and train a CNN using PyTorch. Change the number of layers to investigate how this parameter affects performance. Also, change the activation function to see if you can get better performance. Every time you change the parameters, also change “model\_path” to store your new model.

1. Execute the script to test a CNN implemented by PyTorch:



The script loads the model from ‘models/mnist\_cnn1.pth’ that you have trained in the previous step. Note that you should specify ‘model\_path’ to load the model that you want to test if you have different models.

1. Execute the script and use TensorBoard to visualize training details:



**References:**

[1] https://medium.com/deep-learning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d

-- END --