Georgia Institute of Technology

ECE 8803 HML - Spring 2025

Lab 3A

Due: Sunday, March 16, 2025 @ 11:59 pm EST

Instructions

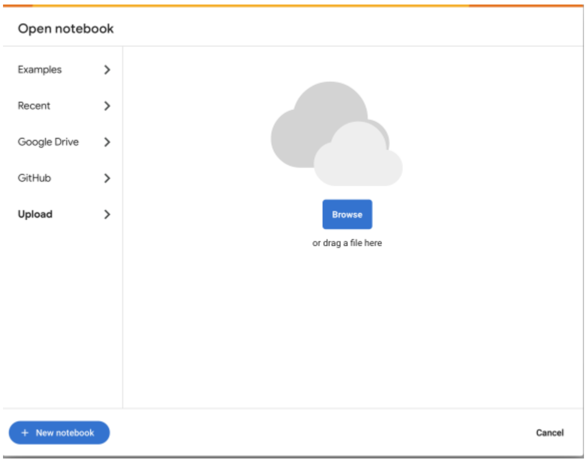
Please read the following instructions carefully.

- The lab is distributed in 2 parts (Each worth 8 points).
- You are encouraged to discuss homework problems with each other, but any copying is strictly prohibited and will be subject to the Georgia Tech Honor Code.
- Late homework is not accepted unless arranged otherwise and in advance.
- For all problems, please post queries on the Piazza. If you add a comment to an answered query, make sure to change the comment to “Unresolved”.

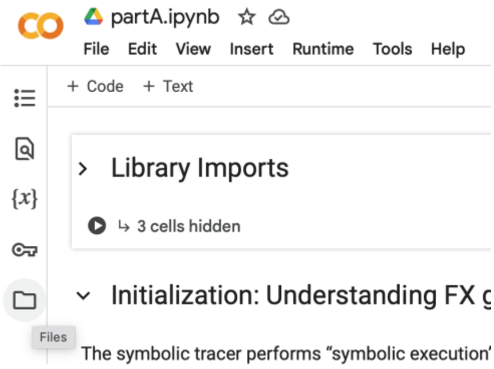
Lab Setup

Part0. Setting up Google Colab

1. Unzip lab3A.zip
2. Go to: https://colab.research.google.com/ [Recommended Browser: Google Chrome]
3. Upload PartA.ipynb



4. Connect with CPU runtime.
5. Goto *Files* (Left Most bar)



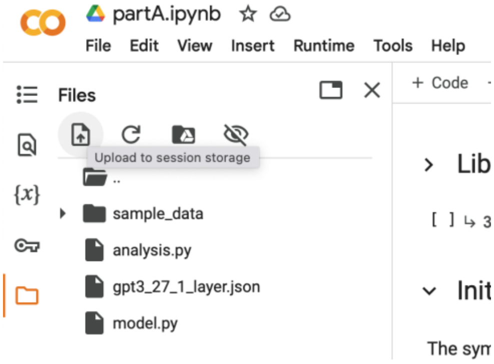
Library Imports

↳ 3 cells hidden

Initialization: Understanding FX g

The symbolic tracer performs “symbolic execution’

6. Select *‘Upload to session storage’*.



7. Upload rest 3 files: *analysis.py*, *model.py* and *gpt3_27_1_layer.json.*

Note: **These 3 files will be deleted every time the runtime is terminated. Remember to periodically download them locally.** Alternatively, you can save them in your Google Drive and load them there.

Lab Description

Before starting the lab, run the 'Library Imports' cell. Installation of required libraries may take 5–10 minutes.

Initialization: Understanding FX Graphs

Read through the provided code for basic understanding of what FX graph is.

A1: Graph Manipulation [1 point]

Modify the FX graph generated in the previous step by replacing a node’s target function.

TASK: Complete the transform function to modify existing graph nodes by replaceing all nodes using the torch.mul operator with the torch.div operator.

A2: Graph Analysis I. [6 points]

Next, analyze a trace graph of a single layer of the GPT-3 2.7B model.

TASK: Review the analysis.py file and complete the following:

1. Set node.shape in NodeProp

- Assign node.shape to the output shape of the node.

2. Set node.latency in NodeProp

- Measure the latency of each operator by running it 10 times and averaging the results.

A.3 Graph Analysis II. [1 point]

TASK: Complete the findHeavyOps function in analysis.py to return the top 3 nodes with the highest latency.

Submission Details

Submit the following 4 files.

1. PartA.pdf generated from ipynb. [video Instruction]
2. analysis.py
3. nodes.csv generated in A.2
4. graph.png generated in A.2