Deep Learning Environment Setup Guide

Target Hardware: ASUS TUF F15 (NVIDIA RTX 3050 Laptop GPU)

OS: Windows 10/11 (Native Setup)

Target Software: TensorFlow 2.10 (Last version with native Windows GPU support)

Phase 1: System Preparation

Do these steps once before touching the command line.

1. Install Visual C++ Redistributable

TensorFlow relies on these Microsoft libraries. Without them, you will get "DLL load failed" errors.

- **Download:** Visual Studio 2015, 2017, 2019, and 2022 (x64)
- **Action:** Run the installer and restart your laptop if asked.

2. Update NVIDIA Drivers

Your RTX 3050 needs the latest drivers to communicate with the older CUDA versions we will install later.

- **Go to:** NVIDIA Driver Downloads
- **Search Settings:**
  - **Product Type:** GeForce
  - **Product Series:** GeForce RTX 30 Series **(Notebooks)** *(Important: Select Notebooks)*
  - **Product:** GeForce RTX 3050 Laptop GPU
  - **Operating System:** Windows 10 or 11 (64-bit)
  - **Download Type:** Game Ready Driver (GRD)
- **Action:** Download and install. Select "Express Installation".

3. Install Anaconda

Anaconda manages the complex environments so you don't break your system Python.

- **Download:** Anaconda Individual Edition
- **Action:** Run the installer.
- **Critical Tip:** When prompted, **DO NOT** check the box that says "Add Anaconda to my PATH environment variable". It is red for a reason. Just click "Register Anaconda as my default Python 3.x".

Phase 2: Creating the "Golden" Environment

We will create a specific environment locked to Python 3.9 to ensure maximum stability with

TensorFlow 2.10.

1. Open the **Windows Start Menu**.
2. Search for and open **Anaconda Prompt** (not PowerShell, not CMD).
3. Type the following command to create the environment (press Enter and then y to confirm):
   conda create -n tf_gpu python=3.9

4. Activate the environment:
   conda activate tf_gpu

   (You should see (tf_gpu) appear on the left side of your command line.)

Phase 3: Installing the GPU Stack

We use Conda to install CUDA and cuDNN because it automatically handles the "Path" variables, preventing common errors.

1. Install CUDA 11.2 and cuDNN 8.1:
   Paste this exact command into your prompt:
   conda install -c conda-forge cudatoolkit=11.2 cudnn=8.1.0

2. Install TensorFlow 2.10:
   Now we install the specific version of TensorFlow via pip:
   pip install "tensorflow<2.11"

3. Install Data Science Tools (Optional but Recommended):
   Install Jupyter, Pandas, and Matplotlib so you can actually work with data:
   pip install pandas matplotlib seaborn scikit-learn jupyter notebook

Phase 4: Verification

Let's confirm the RTX 3050 is recognized.

1. In the same Anaconda Prompt window (ensure (tf_gpu) is still active), type:
   python

2. You will see the Python >>> prompt. Type these lines one by one:
   import tensorflow as tf
   print("Num GPUs Available: ", len(tf.config.list_physical_devices('GPU')))
   print(tf.config.list_physical_devices('GPU'))

3. **Success Criteria:**
   - It should print Num GPUs Available: 1
   - It should list your device: [PhysicalDevice(name='/physical_device:GPU:0',

device_type='GPU')]

4. Type exit() to close Python.

Phase 5: Daily Usage Routine

Whenever you want to work on a Deep Learning project, follow this simple routine:

1. Open **Anaconda Prompt**.
2. Activate your GPU environment:
   conda activate tf_gpu

3. Navigate to your project folder (optional):
   cd Documents/MyProject

4. Launch Jupyter Notebook:
   jupyter notebook

Troubleshooting

- **"DLL load failed":** You likely missed Step 1 (Visual C++ Redistributable). Install it and restart.
- **TensorFlow not finding GPU:** Ensure you did **not** install TensorFlow 2.11, 2.12, or newer. Run pip show tensorflow to check. If it says 2.11+, run pip uninstall tensorflow and reinstall tensorflow<2.11.
- **Kernel