Instructions for setting up Tensorflow (+ dependencies) with GPU acceleration enabled

Bindi M. Nagda

1. Install the latest NVIDIA graphics driver for the GPU on your system
   - For CUDA Toolkit/NVIDIA driver compatibility check: Release Notes :: CUDA Toolkit Documentation (nvidia.com)
2. Install Anaconda
3. Launch Anaconda Prompt
   - In the terminal create a new conda environment: conda create --name tf-gpu
   - Activate the environment: conda activate tf-gpu
   - Begin by installing tensorflow-gpu: conda install tensorflow-gpu
     - This should install the latest tensorflow-gpu available in anaconda. At the time of writing this, it was version 2.6.
     - Conda will also handle installation of other packages
     - Conda will automatically handle the installation of the correct versions of CUDAToolkit and CUDnn
     - To check the packages installed in this environment, type: conda list
   - Then install Keras: conda install keras=2.6
     - The version number to install should be the same as the version number of tensorflow-gpu
   - Finally install other packages you may need. For example:
     - pip install sklearn matplotlib opencv-python sklearn imutils progressbar
     - conda install jupyterlab

   - To launch a jupyter notebook, type jupyter lab in the terminal
   - To launch a jupyter notebook on a different browser/ workstation, type jupyter lab --ip 0.0.0.0 --no-browser and copy-paste the URL it provides into the desired browser

4. To check that GPU acceleration is turned on, run the following in a new cell in Jupyter notebook:

```
import tensorflow as tf
from tensorflow.python.client import device_lib

numGPUs = len(tf.config.experimental.list_physical_devices('GPU'))
print('Num GPUs Available: ', numGPUs)
if numGPUs > 0:
        print(tf.test.gpu_device_name())
        print(device_lib.list_local_devices()[1].physical_device_desc)
```

5. Begin coding in Python.