Lab_1 CNN with MNIST Using TensorFlow

Table of Content

1. File Transfer between Local Device and Rivanna

We can use globus to upload your notebooks, scripts, datasets or any other files needed to conduct experiments. The set-up is simple, follow the following steps:

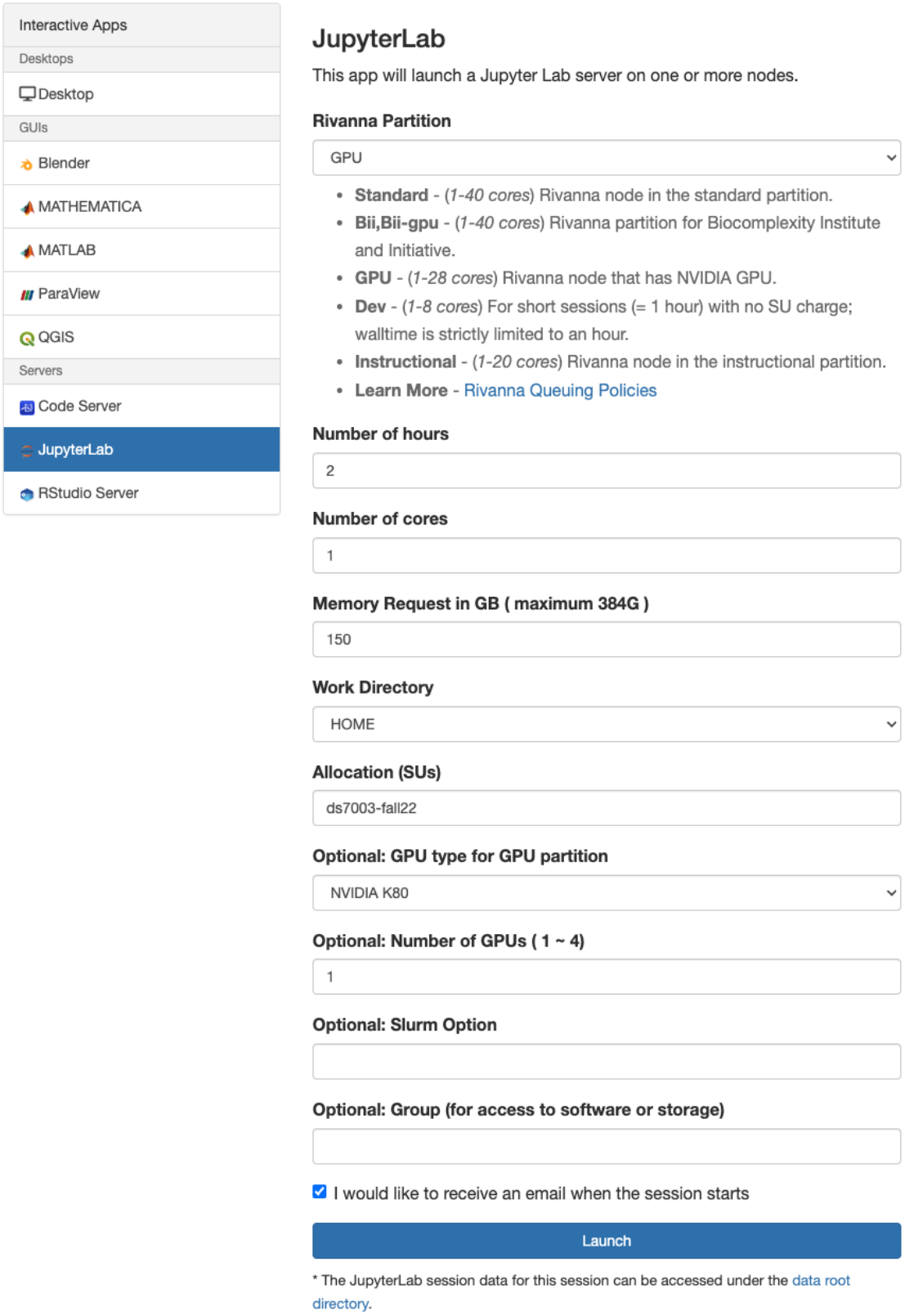
1. Go to globus.
2. Click "Log in".
3. Choose "University of Virginia" from the drop-down list.
4. Start to transfer the files you need.

2. Allocation/Job Request on Rivanna

Next we want to request an allocation using UVA OpenOnDemand. You may use Slurm to set up the allocation and schedule the jobs through an old-school terminal window, if you prefer. However, you could also request an interactive session using OpenOnDemand to run your notebook/scripts. Yes, just like a regular Jupyter Notebook. Let me give you an example:

1. Open OpenOnDemand using the link given above.
2. Click on "My Interactive Sessions" .

3. Click Jupyter Lab, then you will be taken to this page:

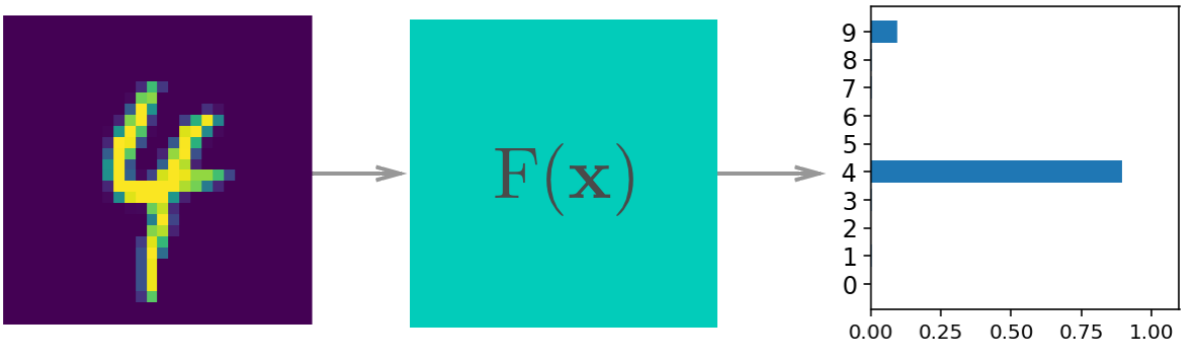


4. Pick the configurations that your job/experiment needs. For demonstration's purposes, we will just use the configurations showing in the figure.
5. Some extra notes:
    1. We have requested the workspace allocation for this class: **ds7003-fall22**.
    2. Normally we would use A100 or V100 GPUs, which requires longer waiting time.
    3. K80 with 2hr should only take a few seconds to get in.

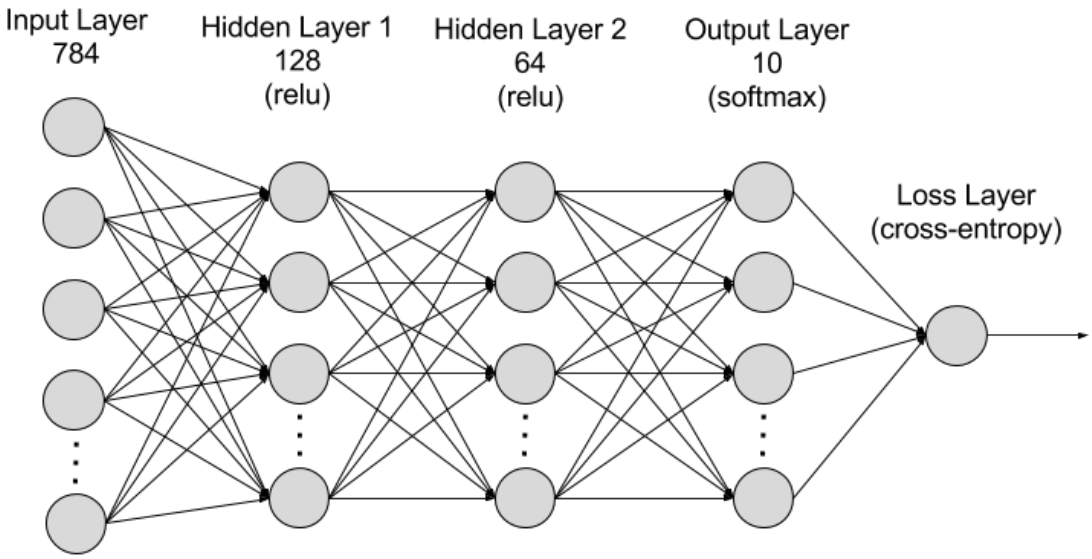
Test with a simple TensorFLow based CNN

1. Now we know how to transfer the files and run them on Rivanna, let's do some simple experiments:
    1. We all know or have heard about the MNIST dataset, which contains greyscale handwritten digits. Each image is 28x28 pixels, you can see a sample below:

    2. Our goal is to build a CNN that can take one of these images and predict the digit in the image:



    3. Here is how the structure for our CNN model looks like:



2. I've a already uploaded a notebook to our Github Repository. Feel free to develop your own network or use the one I provided.
3. Next week, we will implement the same CNN using PyTorch, and do the following:
    1. Compare PyTorch and TensorFlow by computation cost
        1. Execution time of each epoch and total, etc.
        2. Run the experiment on different GPUs and compare the speeds
        3. Discuss on methods to make the experiments run faster.