Turing GPU and Python Environment Tutorial

In this tutorial, the required argument is highlighted with <> for example `<your_username>`. Please replace it with what is said inside, for example, `xzhang17` in this example.

Accessing the WPI network and Turing server.

The first thing you need to do is to have access to the WPI network, either using the WPI network on campus or using VPN (https://hub.wpi.edu/article/444/globalprotect-vpn-client-configuration).

Now, after having the network setup, what you can do now is use ssh

```
1 ssh <your_username>@turing.wpi.edu
```

On Windows, you can access ssh with PowerShell. On Mac OS and Linux, you can use a terminal (zsh, bash, xterm, etc.) You will be prompted to type your password. If this is your first time working with Linux, your password will not be shown. You can just type down your password and press enter when you are done.

First time using Turing.

Ok, after you have done everything. The first thing you need is to import Python and related packages like cudnn. To do this, you can do

```
1 module avail | grep <your_package>
```

To search for the package `your_package` in Turing. The first package you need is Python. Pick one of the python versions you want to use. For the sake of demonstration, let's say we pick `python/3.9.12/uabo2y2 or a newer version`. Now, type

```
1 module load python/3.9.12/uabo2y2
```

to load python. We do the same thing with cudnn and we should be done with importing stuff from Turing.

Create Virtual Environment (venv)

Now, we can install a python virtual environment to enclose the libraries you will use. Type

```
python3 -m venv <your_environment_name>
source <your_environment_name>/bin/activate
```

to initiate your virtual environment. Then, install all the libraries you need.

Create Virtual Environment (conda)

If you are more familiar with conda or you have some trouble with venv, you can download the conda to the server and install it

```
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
bash Miniconda3-latest-Linux-x86_64.sh
```

After the installation is done, restart your bash by typing `bash` if needed. conda will create an environment named "base" for you.

Queuing your job on Turing

Now, after installing everything and you have your code ready to be run on the server. There are two ways to use GPU resources for Turing

- `sinteractive` (required that your terminal is active all the time) Use this for testing your code so that it is runnable
- `sbatch` (running in the background does not require your terminal to active all the time) Use this for training and testing your model

The first one will ask you the resources you need, and you can use that for testing that your code is runnable. To run your code, given that you are working in `sinteractive` is to just run

```
python3 <your_py_code>
```

However, as noted that this required you to have the terminal active all the time. That means if you are running this and sleep your computer or have lost the connection to the WPI network, you will not be able to get any output or recover your run.

Another method to run this is to use `sbatch` to queue your work and let the server do the job for you. Please review the documentation here (https://arc.wpi.edu/slurm-commands/). To do this, you first need to initiate the script with

```
vim run.sh
```

Then, paste this code

```
#!/bin/bash
```

```
#SBATCH -N 1
#SBATCH -n <number_of_cpu_cores>
#SBATCH --mem <memory_uses_in_MB>
#SBATCH --gres=gpu:<number_of_GPU>
#SBATCH -o <stdout_file>
#SBATCH -t <time_in_HH:MM:SS>

srun -l python3 <your_file_name>.py
```

to your script file. Then, post your job on `sbatch` with

```
sbatch run.sh
```

This will give you the jobid of which you can use to look into your queue with

```
squeue -j <your_jobid>
```

and it should show you the queue. If it is done executing, the queue will be empty.

FAQ

1. How can I code on the server?

You can use the built-in text editor vim (vi) in the server. This will have all the basic things you need. However, **we suggest you use VSCode** (https://code.visualstudio.com) with ssh extension installed to code on the server, as your code is relatively large.

2. I can't initiate venv because I don't have pip.

You need to install pip yourself in this case. Please look at https://pip.pypa.io/en/stable/installation/ Links to an external site.for how you can install pip locally.

3. I have my code on my computer; how can I upload it to the server?

You can use scp to upload your file. Create a new shell (i.e., another window for terminal/powershell/cmd) and type

```
scp <your_file> <your_username>@turing.wpi.edu:/home/<your_username>/
```

Or you can use WinSCP for Windows, which offers a GUI version of this command.

The alternative to this is to upload it on an Online service like GitHub, Google Drive, etc. and download it with their command. For example, GitHub will use git, Google Drive will

use gdown (can be installed with pip), and if you have direct access to the file (like long_url/your_file.zip), then you can use wget.

Here is the example script for running. This script will allocate 4 CPU cores, 8GB of RAM, and 1 GPU (A100) for 3 hours to run the python script.

```
#!/usr/bin/env bash
#SBATCH -N 1
#SBATCH -c 4
#SBATCH --gres=gpu:1
#SBATCH -C A100
#SBATCH -t 3:00:00
#SBATCH --mem 8G

python main.py --train_dqn
```