

Running a Colab notebook on our private GPU instance

Fraida Fund

These notes describe how you can connect your Colab notebook to a “private” GPU instance that is hosted specifically for this course.

Prerequisites

You will need an SSH client.

- On Mac or Linux, you can use the default `Terminal` which already has a built-in `ssh` command.
- On Windows, you can download [cmdr](#) and run the `ssh` command inside a `cmdr` terminal.

You will also need two “private” pieces of information:

- The IP address of the GPU instance
- and the key you can use to access it

These will be in the GPU reservation calendar.

Save the private key in a plaintext file with the name `id_rsa_chameleon` (no file extension!) on your laptop.

Your private key must have appropriate permissions set -

- On Mac or Linux, open the terminal in the same directory that you have saved the key in, then run

```
chmod 600 id_rsa_chameleon
```

- On Windows, you can follow [these instructions](#) to set the key permissions on `id_rsa_chameleon` using the GUI.

Save the public key in a plaintext file in the same directory with the name `id_rsa_chameleon.pub`.

Reserve GPU time

A separate calendar link is provided for reserving GPU time.

Please consider these guidelines when reserving GPU time:

- Most students will not need extra GPU time! You should be able to complete all of the work in this course without exceeding the “free” GPU time in Google Colab, unless you are *also* using Google Colab for other projects outside of this course. If you *can* complete the work using Colab’s free hosted runtime, you should.
- You may reserve time at least 2 hours, and up to 3 days, in advance.
- You may not reserve more than 1 hour per day.

You should make sure that your code is “ready to run” before the beginning of your reserved GPU time. Then you can use your reserved time to just run your notebook from beginning to end, and save the results.

Connecting to the GPU instance

At your reserved time (not earlier!), run
where

- in place of `IP_ADDRESS`, substitute the IP address of the GPU instance.
- in place of `PORT` use: 50000 + the numeric part of your net ID. For example, if my net ID is `ab123`, I will use: 50123.

This will set up a tunnel between your local system and the GPU instance. Leave this SSH session running.
Inside the SSH session, run:

(again substituting your `PORT` number) and leave it running.

In the output of the cell above, look for a URL in this format, with the word “localhost” in it:

Copy this URL - you will need it in the next step.

Now, you can open Colab in a browser. Click on the drop-down menu for “Connect” in the top right and select “Connect to a local runtime”. Paste the URL you copied earlier into the space and click “Connect”. Your notebook should now be running on our GPU instance.

When you are finished

When you are finished working OR at the end of your reserved time slot, whichever comes first -

- use `Ctrl+C` in your terminal session to stop the Jupyter notebook server
- when prompted, type `y` and hit Enter to confirm
- type `exit` and hit Enter to close the SSH tunnel

Addressing common problems

“Could not request local forwarding” Q: When I use the SSH command
it says:

A: This will happen if you already have something else running on this port on your laptop. You can change the port in the command, e.g. use 51000 + numeric part of net ID.

Warnings when using Tensorflow When I import `tensorflow`, I see the following warnings:

and when I try to `fit` a model, I see:

Is this bad?

This server has different library versions and a different GPU type than the Colab hosted runtime, so you may see some warnings/notifications that you wouldn’t see in Colab. It’s not a cause for concern, and you can safely ignore the notifications shown above.