Guidelines: Setting up Your Machines for Your Coursework

Introduction to Machine Learning (Autumn Term)

1 Overview

You should implement all code for both courseworks in **Python 3**.

For coursework 1, you should assume that your code will be manually run on a DoC lab machine by the person assessing your work.

For coursework 2, your code will be assessed and tested on `LabTS` (https://teaching.doc.ic.ac.uk/labts/). You will be given a GitLab repository for your group with the skeleton code for each coursework. The link to this will be available on `LabTS`. If you are not familiar with Git, you can find a quick introduction in the GitHub documentation: https://guides.github.com/introduction/git-handbook/.

2 Python

Both `LabTS` and the Ubuntu workstations in the DoC labs run Python version 3.10. Therefore, please make sure your code runs on this version of Python.

2.1 Working on DoC lab workstations

On a DoC machine, you may use the following Python virtual environment to run your code. This virtual environment contains all the necessary packages for both courseworks, and closely mimics the one on `LabTS`.

```
$ source /vol/lab/intro2ml/venv/bin/activate
(venv) $ python3 -c "import numpy as np; import torch; print(np); print(torch)"
<module 'numpy' from '/vol/lab/intro2ml/venv/lib/python3.10/site-packages/numpy/__init__.py'>
<module 'torch' from '/vol/lab/intro2ml/venv/lib/python3.10/site-packages/torch/__init__.py'>
```

You can connect to the lab machines from home using SSH. You can find the list at https://www.imperial.ac.uk/computing/csg/facilities/lab/workstations/.

2.2 Working on your own system

You can also create your own virtual environment locally on your own machine. A `requirements.txt` file is provided on Scientia, and also with your skeleton code (for coursework 2). This is the same `requirements.txt` file used to create the `intro2ml/venv` above. Use this `requirements.txt` file to install the appropriate packages.

The following example assumes that you are using a Unix based machine.

```
$ cd your_project_directory
$ python3 -m venv venv
$ source venv/bin/activate
(venv) $ pip install --upgrade pip
```

```
(venv) $ pip install -r requirements.txt
(venv) $ python3 -c "import numpy as np; import torch; print(np); print(torch)"
<module 'numpy' from '/vol/lab/intro2ml/venv/lib/python3.10/site-packages/numpy/__init__.py'>
<module 'torch' from '/vol/lab/intro2ml/venv/lib/python3.10/site-packages/torch/__init__.py'>
```

Please do NOT include the generated `venv` directory in your submission. For coursework 1, please do not include it in your ZIP file. For coursework 2, please do not add/commit the directory to your gitlab repository. You can add the directory to the `.gitignore` file in the repository (we have already added `venv`, so you do not have to do this unless you named your virtual environment differently).

3 LabTS (For coursework 2)

For coursework 2, we will provide a suite of tests for you to test your code prior to submission using `LabTS`.

These 'public' tests are purely to ensure that your code runs correctly in the `LabTS` environment. They are **NOT** aimed to provide you with an extended testing environment. Thus, do not expect them to give you detailed error messages. You should also double-check components or functions that are not directly tested by them.

You are responsible for testing your own code offline, including components or functions that are not directly tested by these tests. The available tests are also only examples; the final assessment will be performed using a different test suite not available to you.

The tests may also evolve between release and the coursework deadlines.

To test your code, push to your GitLab group repository and request the tests through `LabTS`. You will receive a PDF report showing which tests you passed or failed after they have finished running.

We highly recommend that you test your code on `LabTS` as soon as possible and throughout the project, to avoid discovering incompatibility issues with the test environment at the last minute. You can test your code on `LabTS` as often as you need to, and we will consider for marking only the commit corresponding to the **SHA1 token** you submitted on CATe.

4 Your responsibility

It is your own responsibility to ensure that your code runs on the DoC lab machines (for coursework 1) and on `LabTS` (for coursework 2).

We reserve the right to reduce your marks by 30% for any bits of your code that cannot be run.