Deep learning practical 1

Jupyter & colab setup

Practical overview

Welcome to the first deep learning practical. This is a very light setup practical, with the following purpose:

- Ensure a suitable deep learning training environment.
- Chat with the staff about any questions about the module.

Environment setup

It is important that, after this practical, everyone has an environment setup. There are generally four ways to run the code and train the large deep learning models required for this module:

- Use Google Colab—easy to get started, but the usage gets throttled and the kernel is not persistent.
- Use the NCC Jupyter server—recommended for most students. The I/O can be a little slow.
- SSH to NCC and use SLURM.
- Use the lab machines or your home machines—not recommended if the GPUs are inferior.

Using Google Colab

- You need a Google account for this
- Simply open an existing .ipynb in Colab and run the cells—here are two to get started with:
  - Train a classifier: download
  - Train a convnet: download
- Check that your 'Runtime > Change runtime type' is a GPU

NCC usage via Jupyter (recommended)

- Connect to a machine from the university network, otherwise use the VPN to connect from outside
- Login to `ncc1.clients.dur.ac.uk/COMP0000` with your CIS username and password (if you get an authentication failure, you might not have an account on NCC and need to email Rob Powell)
- Select appropriate preferences for the job, e.g. `ug-gpu-small` with `1 GPU`
- Once it takes you to your home directory click "Assignments" tab and "Fetch" the environment creation folder
- Click the title of the folder "Create Jupyter Environment Script" and click the "CreateJupyterPythonEnvironment" script that is inside it
- In the second code cell you need to enter a Python version, and two names (these can be the same if you like). The virtualEnvName becomes the directory name of the virtualenv in your NCC home directory. The jupKernelName is what appears in the Jupyter menu later. If you're only using PyTorch you shouldn't need modules as CUDA is bundled
- Run the whole notebook (Kernel > Restart and Run All). Once it has finished (about 4-5 mins) you can close this tab and return to the previous tab that should still be on the Assignments page.
- Click "Files" to see home directory again and then refresh the page otherwise the new kernel won't appear in the list!
- Click "New" and select your kernel name from the dropdown list
- Run your `!pip install` commands (only needs to be done the first time you ever use the kernel, unlike Colab, as our storage is persistent)
- Download, upload to NCC and run the classifier.ipynb and convnet.ipynb examples

NCC usage via SSH

Only do this if you know what you're doing, e.g. have used SLURM before:

- SSH to the head node `CIS-username@ncc1.clients.dur.ac.uk` (either via VPN or Mira)
- Do not run code on this head node. Doing so will cause problems for all other users, can interrupt active postgraduate research and may get you banned
- Before running any code, read the NCC documentation carefully: `ncc1.clients.dur.ac.uk` and do your compute with SLURM
- We recommend using `tmux` for a persistent shell along with `zsh`
- Please respect other users in the queue

PyTorch (and optional JAX) preliminaries

If you have successfully set up your environment, you are welcome to leave this first practical early. However, we now recommend you take the time to follow the PyTorch and (optional) JAX preliminaries from the "Dive into Deep Learning" book, using your newly setup NCC environment:

`https://d2l.ai/chapter_preliminaries/ndarray.html`

- Work your way from section 2.1 (Data Manipulation) to the end of section 2.3 (Linear Algebra). If you have any questions, please ask the instructors.
- We would recommend reading ahead in this book and practicing the examples in both PyTorch and JAX. How does JAX differ from PyTorch in these examples?
- You may wish to also take time now to integrate your workflow with `vscode` or cursor, and discuss best/recent practices with the instructors and those around you.
