Deep Learning

Yingzheng Li, Harry Coppock, and Bernhard Kainz

Delivery team 2024

Bernhard Kainz, part I

Harry Coppock, coursework, lectures

Yingzheng Li, part II

GTAs:

- Adam Dejl (ad5518)
- Arvin Lin (al4419)
- Caner Korkmaz (ck223)
- Carles Balsells Rodas (cb221)
- Hadrien Reynaud (hjr119)
- Junqi Jiang (jj1820)
- Konstantinos Barmpas (kb1716)
- Lapo Rastrelli (lr4617)
- Liu Li (ll1320)
- Matthew Baugh (mb4617)
- Maxence Faldor (mf1022)
- Myles Foley (mlf20)
- Nikolai Rozanov (nr1713)
- Osama Hassan (oh1117)
- Sarah Cechnicka (sc7718)
- Weitong Zhang (wz1820)
- Xavier Sumba Toral (xxs22)
- Xiang Yin (xy620)
- Xiaotong Ji (xj15)
- Yiming Luo (yl10621)
- Zijing Ou (zo122)

Learning outcomes

- After this course you will know a little bit more about:
  - Feature extraction, convolutions and CNNs
  - Common Network architectures, losses
  - Parameter optimisation

  How it works

  - RNNs, LSTMs, GRUs
  - VAEs and GANs
  - Attention and Transformers

  Why it works

  - Deep learning programming frameworks
  - Applications of deep learning
  - Using GPU resources

  How to do it

Good to know

70015 Mathematics for ML (recommended)
60012 Introduction to ML (soft prerequisite, please read the basic ML notes if you haven't done this course)
60006 Computer vision
zz70014 ML for imaging
70016 Natural language processing

70028 Reinforcement learning

Beyond: 70887W Deep Graph-Based Learning

What we expect you to know already

- Perceptrons, Sigmoid neurons
- Feed-forward neural networks
- Computational graph, Chain rule and backpropagation
- Gradient descent and its variants
- Linear Algebra
- Familiarity with Python programming
- Prior exposure to libraries like TensorFlow or PyTorch would be beneficial

Reference

- Dive into Deep Learning https://d2l.ai/
- I. Goodfellow, Y. Bengio, A. Courville, *Deep learning*. MIT Press, 2016 www.deeplearningbook.org
- Some lectures have been heavily influenced by Material from Michael Bronstein, Kilian Weinberger, Stefanos Zafeiriou, Andreas Maier, Alex Smola, Serena Yeung, Fei-Fei Li

Structure

- Lecture – watch videos, repetition and discussion in class
  - in Huxley 308 and Q&A on MS Teams Fri 9-10
  - Post and discuss questions also in advance on EdStem/MS Teams lab queue please
- Tutorials – Q&A sessions with TAs
  - on Teams and Huxley 308 Fri 10-11
  - Lab queue will be attended by GTAs Friday morning
- Coursework – hands-on programming exercises: individual with lab queue on MS Teams

Support structure

Lecture

Coursework

Tutorials

GPUs

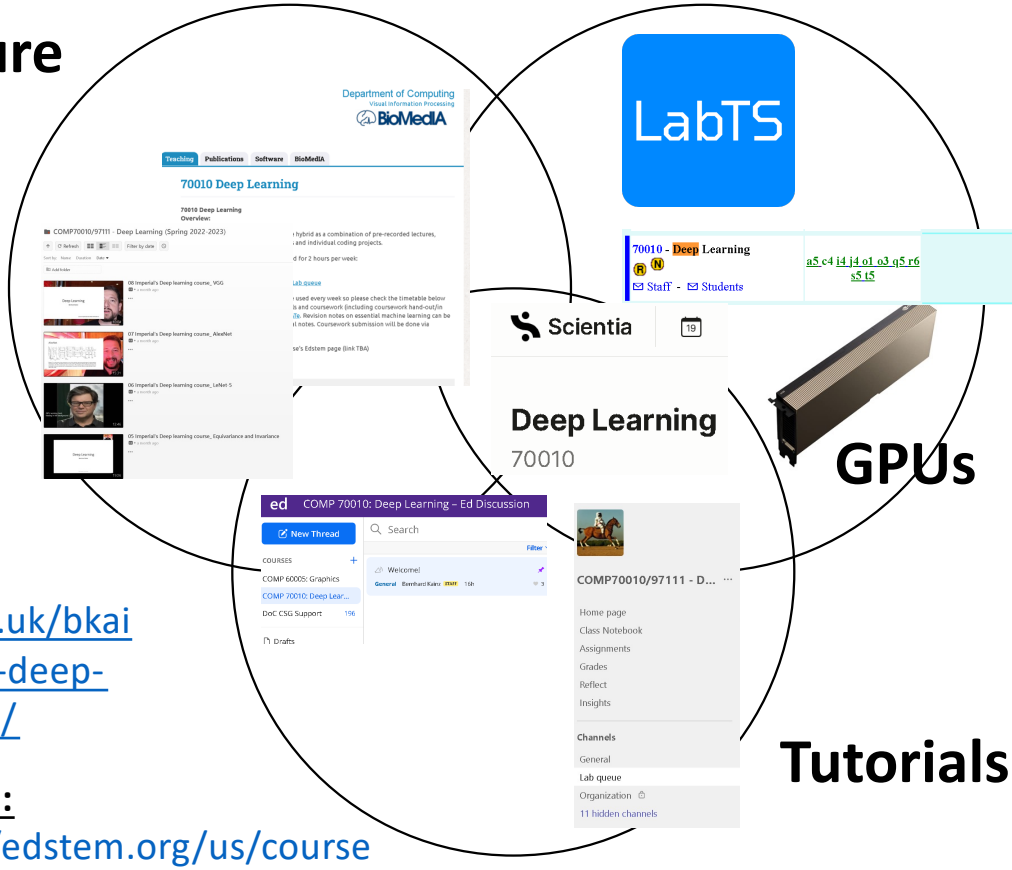
Panopto
https://imperial.cloud.panopto.eu/Panopto/Pages/Sessions/List.aspx#folderID=%2213e9170f-ef01-4d9c-84f6-b02101004540%22

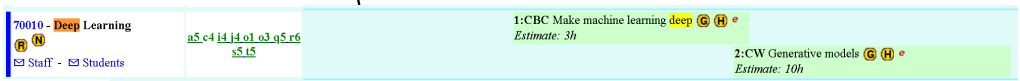
Course website:
http://wp.doc.ic.ac.uk/bkainz/teaching/70010-deep-learning-from-2023/

EdStem:
https://edstem.org/us/courses/46843/discussion/

Scientia:
https://scientia.doc.ic.ac.uk/2324/modules/70010/exercises

MS Teams lab queue
https://teams.microsoft.com/l/channel/19%3a59594c3f5432490e88331b3eef942db8%40thread.tacv2/Lab%2520queue?groupId=f5d4c850-1a83-4969-a72f-32bd6767aff8&tenantId=2b897507-ee8c-4575-830b-4f8267c3d307



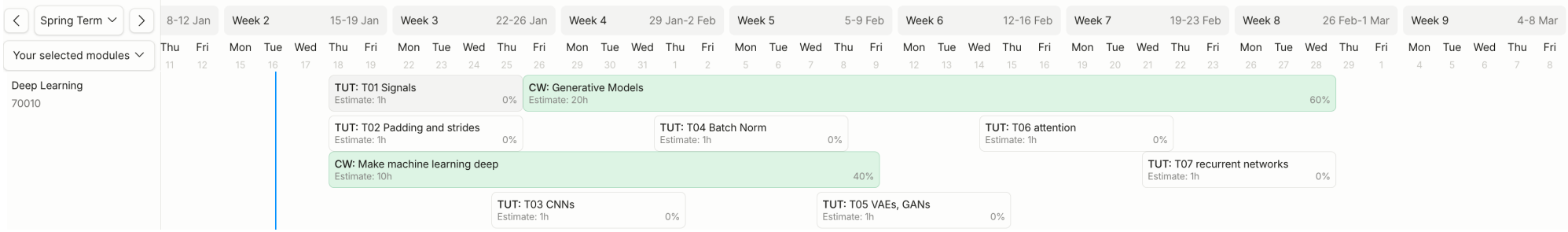


Lab queue for coursework and tutorial

- Fri 10-11 UK time
- Post your question in the lab queue channel.
- Be specific, we will ignore you if you only post ‘I have a question’
- Reason: you will be picked up by one of the GTAs and invited into their break out room. There you can discuss in detail.
- Having a specific initial question allows other students with the same question to join and we can have a more engaging discussion.

Tutorial/coursework

- solve a problem sheet per week (provided on CATe and Scientia)
- not assessed
- help available on the lab queue or in Huxley 308 Fri 10-11 UK time
- solutions provided each following Monday on CATe



40% **60%**

Coursework – using GPU recourses is ILO

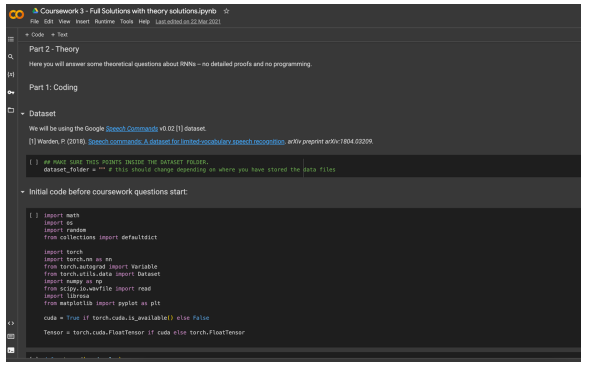
- 2 Tasks, both assessed, Task 1: 40%, Task 2: 60%
- solution two weeks after deadline (to accommodate late submissions)
- results as soon as possible but you are many, which means we'll need some time for marking
- Jupyter notebooks
- Recommendation: use all options.
  - Coursework 1:
    - https://colab.research.google.com/ (dynamic GPU usage limit)
    - https://studiolab.sagemaker.aws/ (4 h per 24 h usage limit)
    - https://www.paperspace.com/
    - Imperial DoC GPU cluster https://www.imperial.ac.uk/computing/people/csg/guides/hpcomputing/gpucluster/
  - Coursework 2:
    - https://www.paperspace.com/ – Harry intro
    - Computing lab GPUs: ssh into machine, if free, train (risky re hard reset)
    - Imperial DoC GPU cluster https://www.imperial.ac.uk/computing/people/csg/guides/hpcomputing/gpucluster/
- Submission via LabTS for coursework 1 and Scientia/emarking for coursework 2

Read the docs!

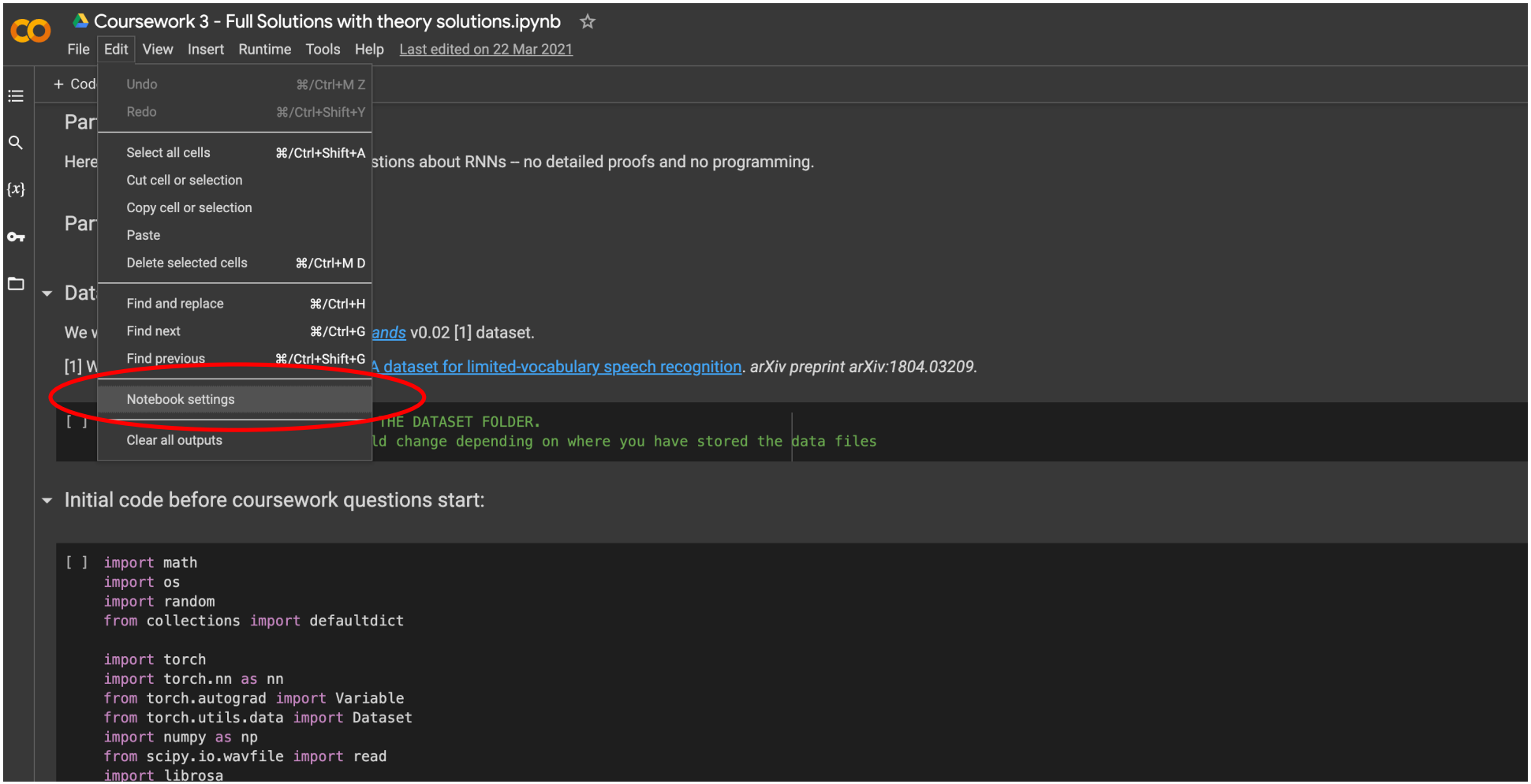
- https://www.imperial.ac.uk/computing/people/csg/guides/hpcomputing/gpucluster/
- https://www.doc.ic.ac.uk/~nuric/teaching/remote-working-for-imperial-computing-students.html
- https://edstem.org/us/courses/15501/discussion/
- https://www.imperial.ac.uk/computing/people/csg/guides/

What is a jupyter notebook?

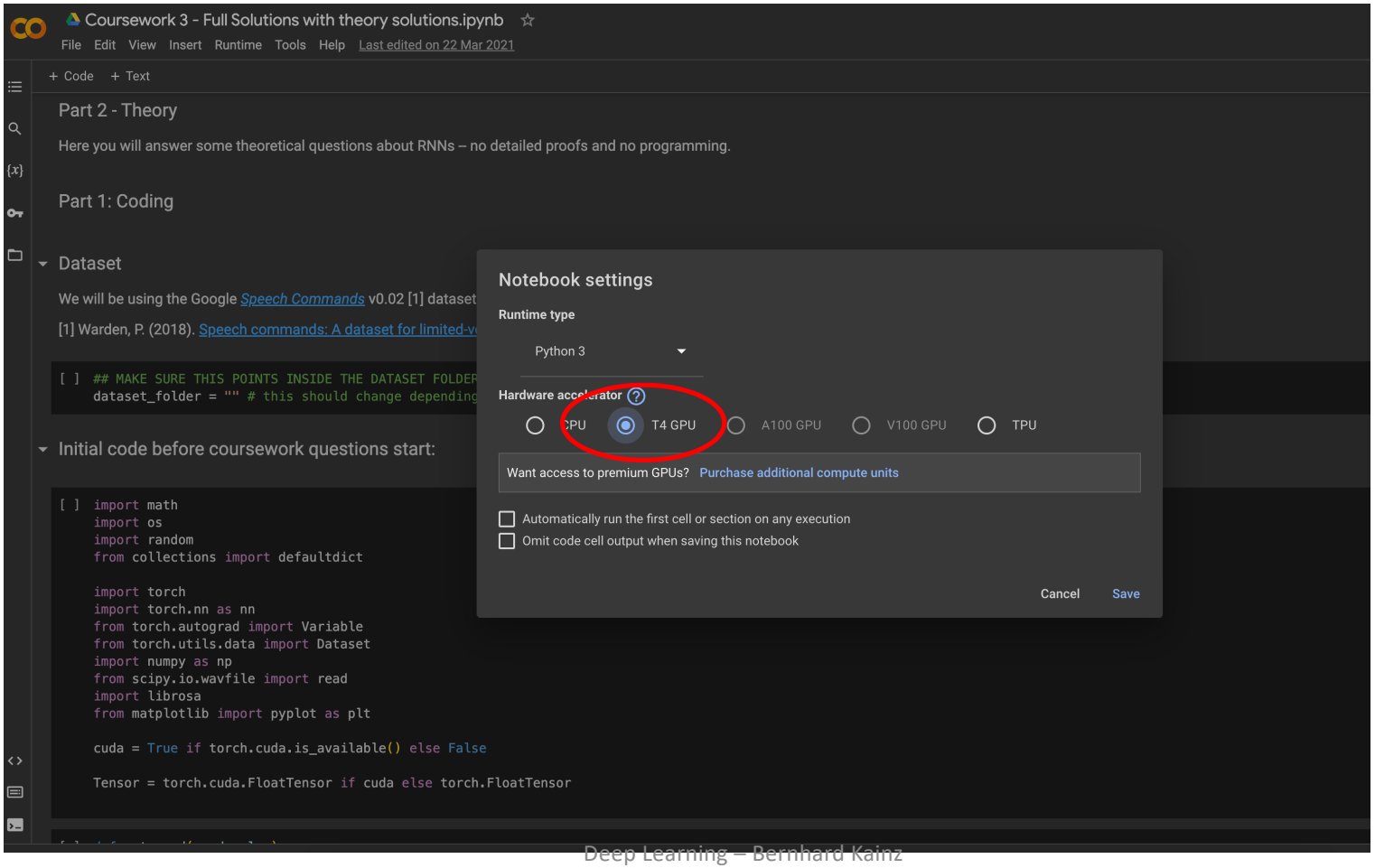
- "A Jupyter Notebook is an open-source web application that allows you to create and share documents containing live code, equations, visualizations, and narrative text."
- Key Features:
  - Interactive coding environment
  - Supports multiple programming languages (Python, R, Julia, etc.)
  - Integration of code and rich text elements (paragraphs, equations, figures, links)



GPU Option 1: Google Colab

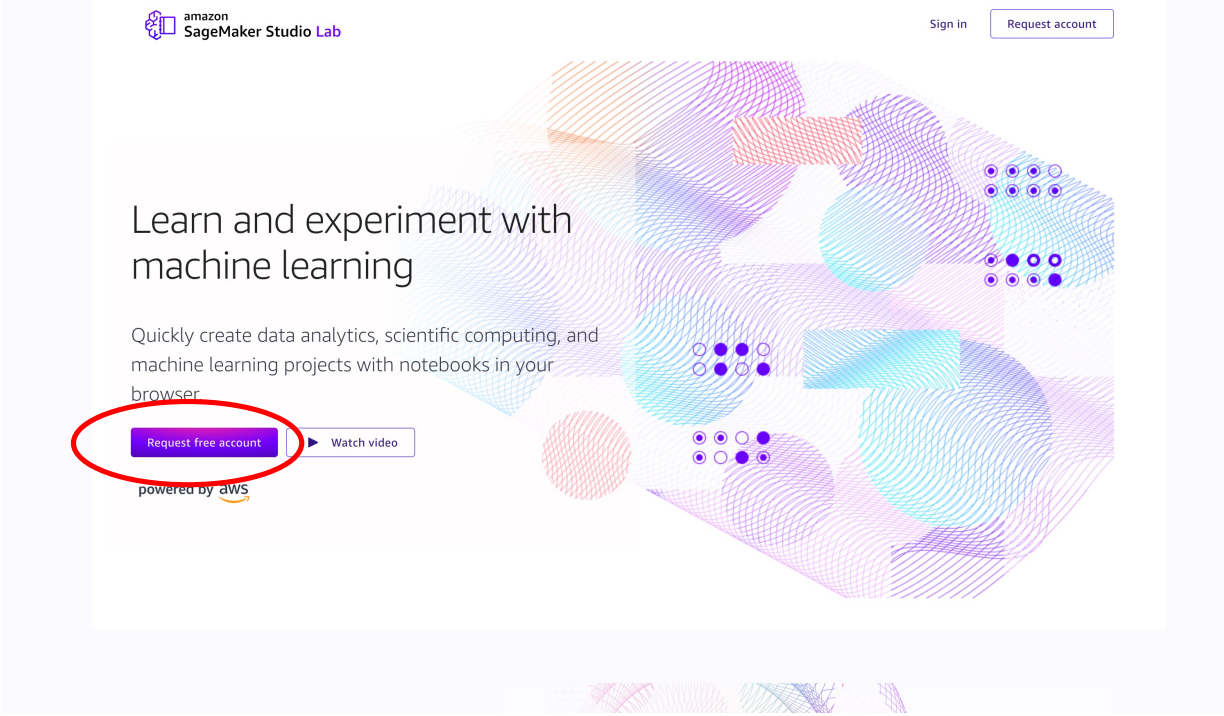


GPU Option 1: Google Colab



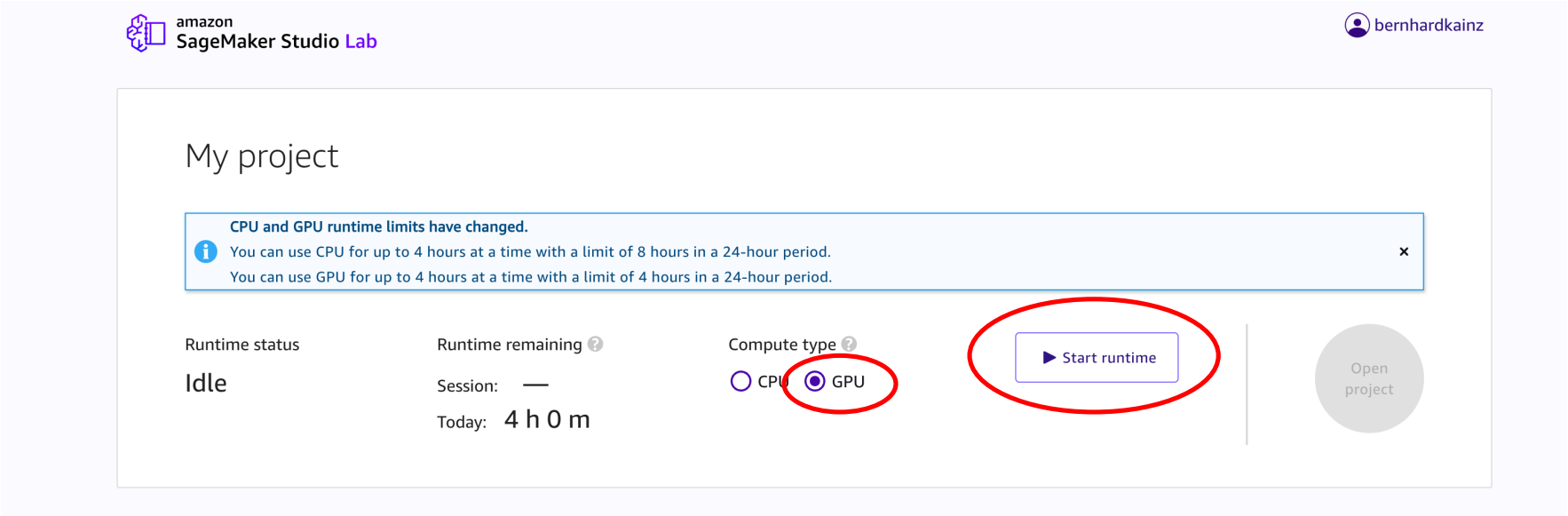
GPU Option 2: sagemaker studiolab

- https://studiolab.sagemaker.aws/ (4h per 24h GPU limit)



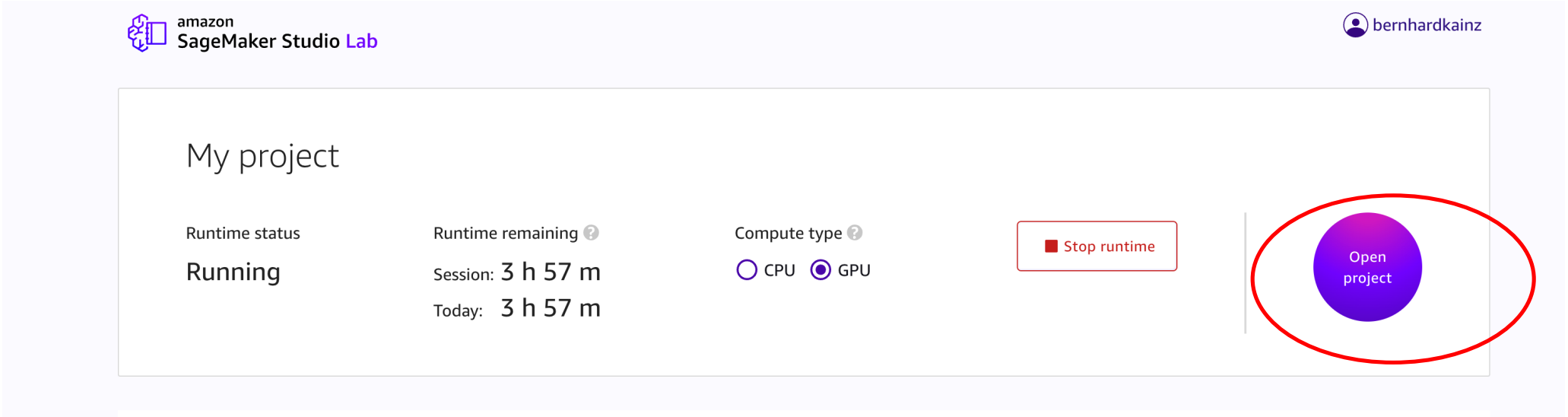
GPU Option 2: sagemaker studiolab

- https://studiolab.sagemaker.aws/



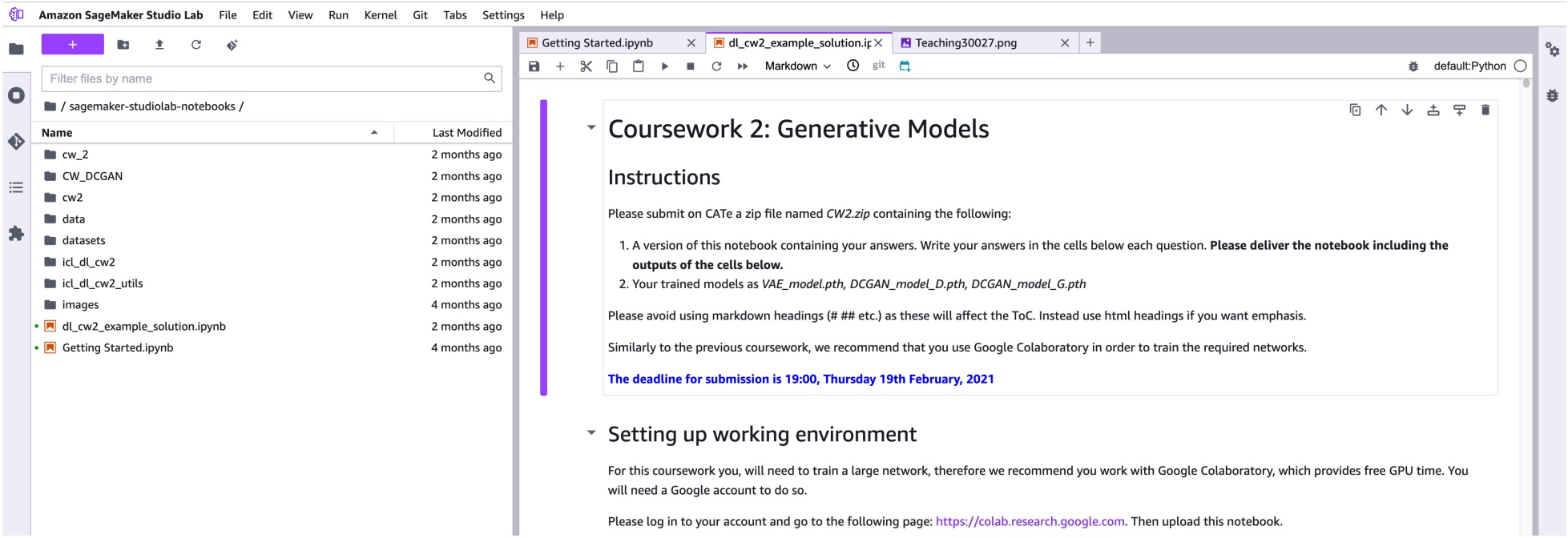
GPU Option 2: sagemaker studiolab

- https://studiolab.sagemaker.aws/



GPU Option 2: sagemaker studiolab

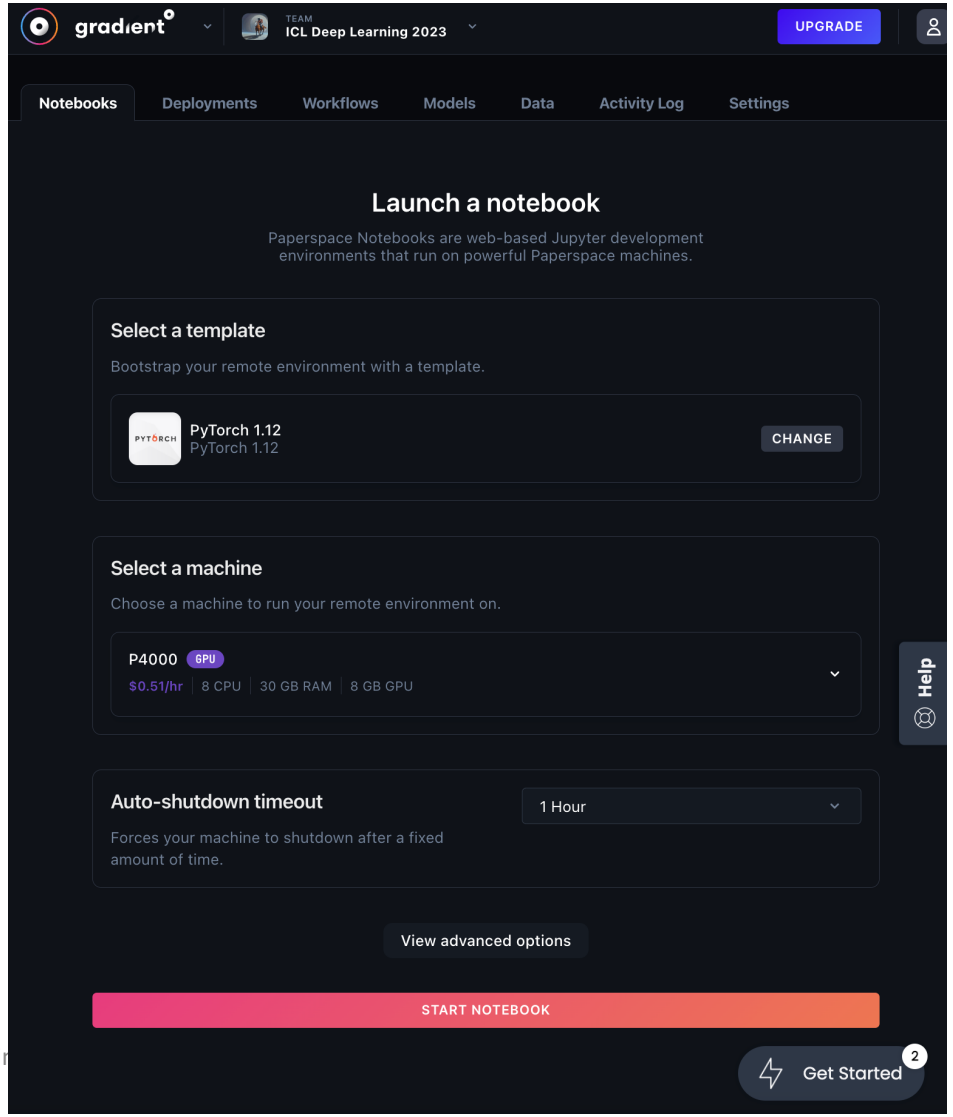
- https://studiolab.sagemaker.aws/



GPU Option 3: paperspace

- https://console.paperspace.com/
- Harry Coppock

- **SHUT DOWN YOUR NOTEBOOK WHEN NOT USED!**
- **SHUT DOWN YOUR NOTEBOOK WHEN NOT USED!**
- **SHUT DOWN YOUR NOTEBOOK WHEN NOT USED!**
- **SHUT DOWN YOUR NOTEBOOK WHEN NOT USED!**
- When the budget is used up, paperspace GPUs will be gone.



Paperspace

1. Write email to dl-paperspace@imperial.ac.uk (goes to us)
2. Get invitation
3. Accept and get started with your notebook

VSCode and paperspace:
https://docs.paperspace.com/gradient/notebooks/notebooks-remote-kernel/

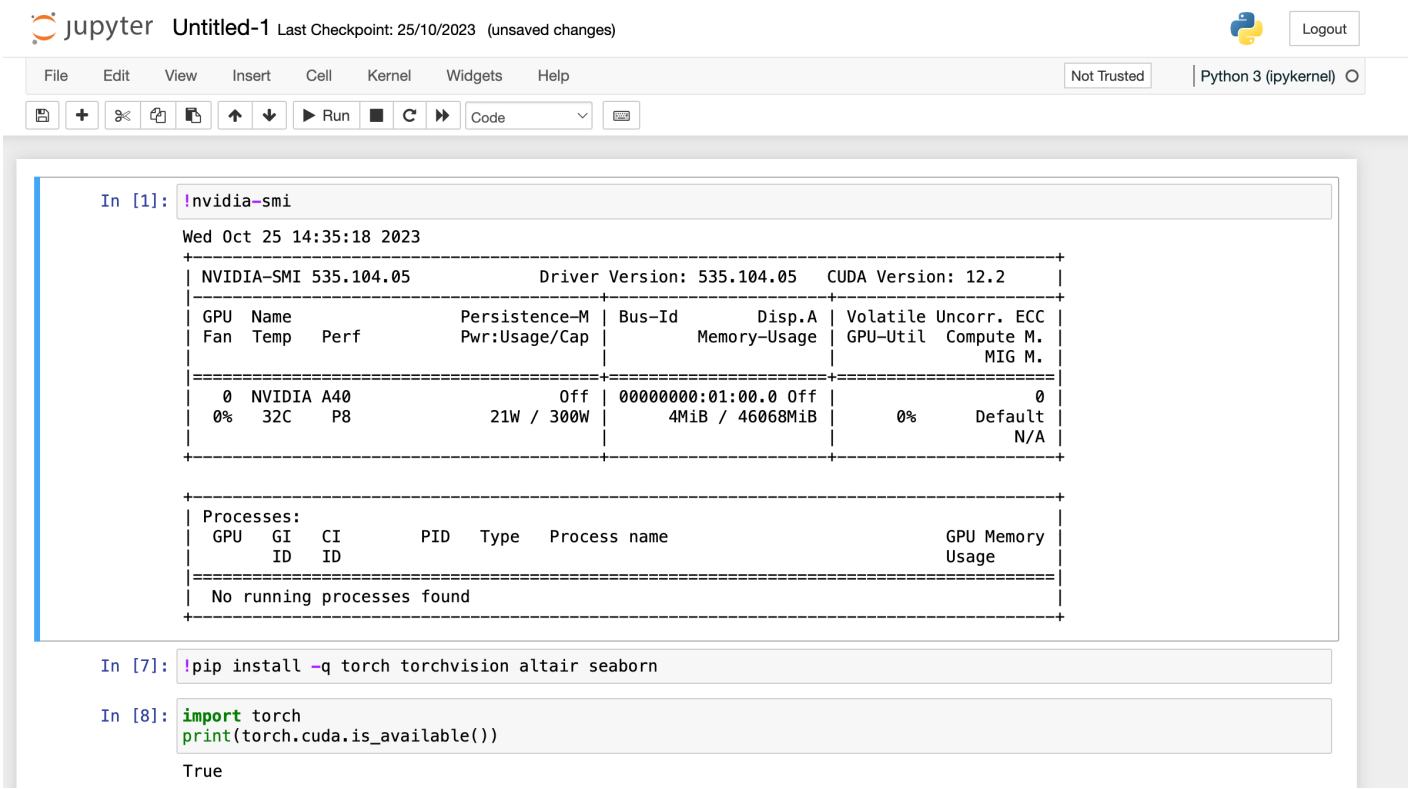
Must spend credits until 01/03/2024 but credits are limited.

GPU Option 4: DoC lab GPUs

- Locate a suitable machine: https://www.imperial.ac.uk/computing/people/csg/facilities/lab/workstations/

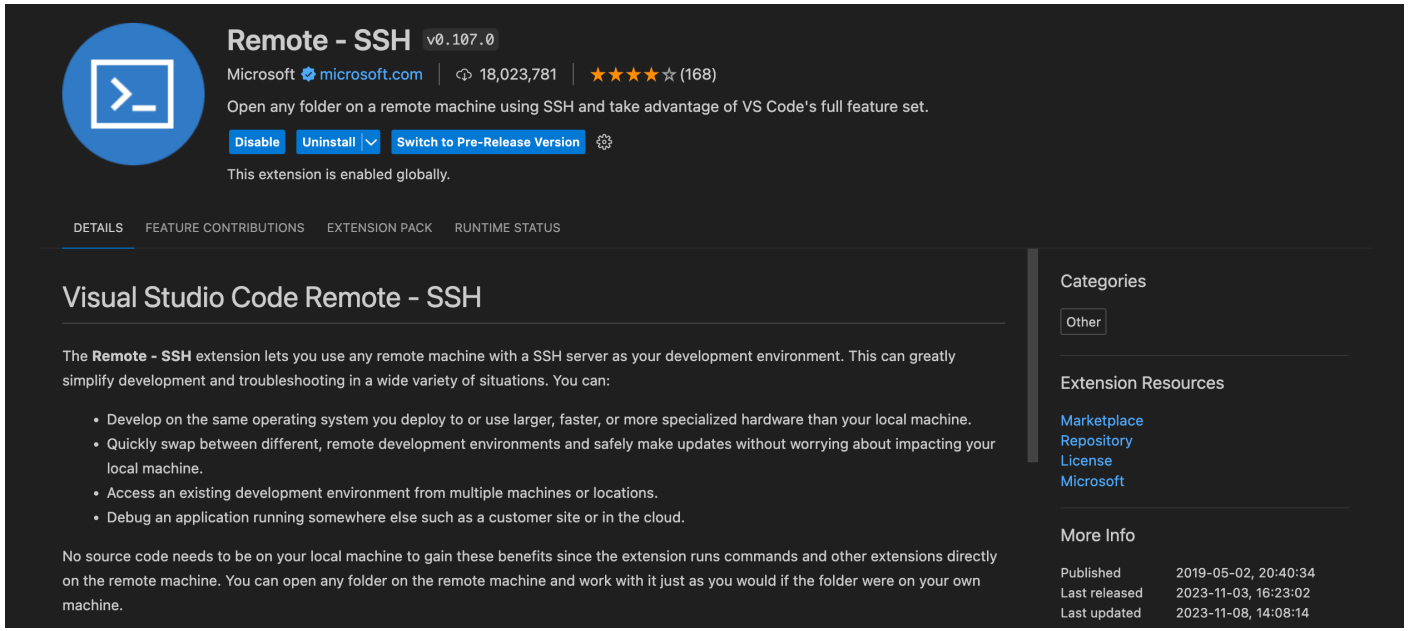
- `ssh -L 8888:localhost:8888 -A -J {USER}shell3.doc.ic.ac.uk \`
  `{USER}@ {labmachine}.doc.ic.ac.uk`
- Check if GPU or machine is busy with `nvidia-smi and top htop`
- (create and activate your own virtual environment as applicable)
- `pip install jupyter`
- `jupyter notebook --no-browser` OR `python -m notebook  --no-browser`
- Open http://127.0.0.1:8888/ in your local browser

GPU Option 4: DoC lab GPUs



VSCode remote

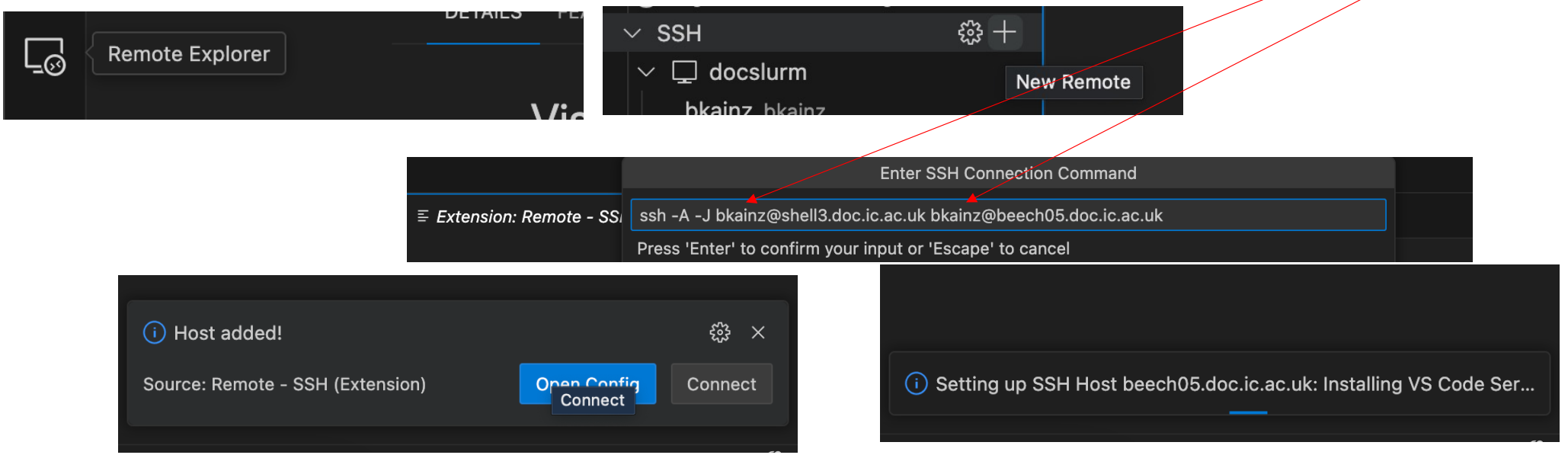
- https://code.visualstudio.com/
- Install remote extension



VSCode remote

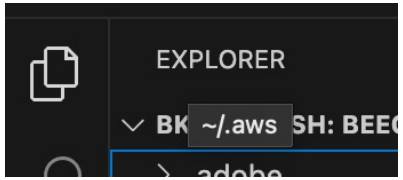
- https://code.visualstudio.com/
- Set up new remote

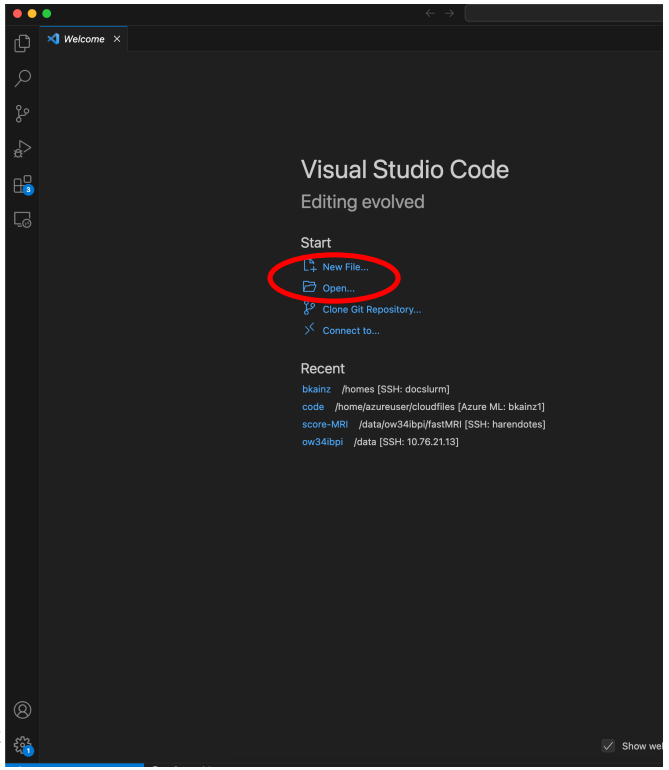
Your username here



VSCode remote

- Open coursework notebook from your home directory directly or through explorer





Imperial DoC GPU cluster

- How to run an interactive notebook on a SLURM cluster?
  - What is SLURM?
  - How to submit a job?
  - How to connect to a SLURM job via ssh?
  - VSCode remote environment example

What is SLURM?

- SLURM (Simple Linux Utility for Resource Management)
- An open-source, fault-tolerant, and highly scalable cluster management and job scheduling system.
- Widely used for high-performance computing (HPC) in both academic and industrial environments.
- Resource Allocation: Efficiently manages resources (CPUs, GPUs, memory) on a computing cluster.
- Job Scheduling: Queues and schedules jobs (tasks) submitted by users, optimizing for resource availability and constraints.
- Scalability: Capable of handling large clusters with thousands of nodes.

GPU Option 5: DoC cluster GPUs

- Users submit jobs using SLURM commands.
- SLURM allocates resources based on job requirements and cluster policies.
- Job execution and monitoring, with users receiving updates on job status.
- https://www.imperial.ac.uk/computing/people/csg/guides/hpcomputing/gpucluster/
- Frontend:
- `ssh gpucluster2` OR `ssh gpucluster3`

Setup

- `cd /vol/bitbucket/{USER}`
- `python -m venv /vol/bitbucket/{USER}/dlenv`
- `source /vol/bitbucket/{USER}/dlenv/bin/activate.csh`
- `pip3 install ipykernel`
- `pip3 install jupyterlab`
- `pip3 install jupyterhub`

GPU Option 5: DoC cluster GPUs TERMINAL 1

- Store datasets/scripts in /vol/bitbucket/**{USER}**
- Add CUDA to your PATH from /vol/cuda/{version}
- Allocate resources from gpucluster2/3:
  - `salloc --gres=gpu:1`

```
salloc: Assigning requested gpu/s:
salloc: gres:gpu:1
salloc: Granted job allocation 97181
salloc: Nodes cloud-vm-42-53 are ready for job
cloud-vm-40-244:~>
```

- Run `squeue` and find your job – **keep terminal open**

```
97114   resgpuD wconf-jo   am8520  R 1-23:11:21      1 cloud-vm-41-147
97127   resgpuD run_rand   el2522  R 1-02:19:53      1 cloud-vm-41-147
97172   resgpuD wconf-jo   am8520  R   10:56:28      1 cloud-vm-40-226
97180   resgpuD test_lla    gif22  R      25:03      1 cloud-vm-42-53
97181   resgpuD interact   bkainz  R       0:50      1 cloud-vm-42-53
```

GPU Option 5: DoC cluster GPUs **TERMINAL 2**

- ssh -t -L 10001:localhost:10001 **{USER}**@shell3.doc.ic.ac.uk "/vol/linux/bin/slurm_sshtojob.sh -g -w ~/ -p 10001 -e /vol/bitbucket/**{USER}**/dlenv"
- Open http://localhost:10001/lab?token={from terminal} in your browser
- Open DL coursework notebook from your home directory
- scancel <your job ID> (or close TERMINAL 1) **when done!**

DoC Gpucluster with VSCode

- Remote extension as above
- `gpucluster2` or `gpucluster3`
  - `salloc --gres=gpu:1`
  - `squeue` to check your job and see the node

```
             97479   resgpuD mut_info   tfb115  R      36:37      1 cloud-vm-40-226
             97480   resgpuD mut_info   tfb115  R      36:33      1 cloud-vm-42-53
             97486   tempgpu     wrap   bkainz  R       0:05      1 kingfisher
cloud-vm-40-244:~>
```

- VSCode:
  - `ssh -A -J {USER}@shell3.doc.ic.ac.uk {USER}@kingfisher.doc.ic.ac.uk`

Read the docs!

- https://www.imperial.ac.uk/computing/people/csg/guides/hpcomputing/gpucluster/
- https://www.doc.ic.ac.uk/~nuric/teaching/remote-working-for-imperial-computing-students.html
- https://edstem.org/us/courses/15501/discussion/
- https://www.imperial.ac.uk/computing/people/csg/guides/

In case of problems

- Read the docs!
- Adjust, try again
- Read the docs!
- Adjust, try again
- Ask on Ed or MS-teams lab queue
- Read the docs, if you find a solution, post the answer on Ed or lab queue

changes to address student feedback from previous years

- more available GPUs – made a deal with paperspace in 2021, google colab has a time-limit on GPU use and does not want to let us pay for your use in bulk. Added 100 GPUs in local DoC Imperial infrastructure in October 2023.
- too much coursework: dropped coursework 3 (RNNs) this term, also too much overlap with NLP course
- more time for coursework 2, which is tempting to play around with it
- overall reduced coursework workload
- Lecture is structured on purpose into to points-of-view: One Engineering-focused view covering fundamental techniques and approaches and one mathematics-focused part, covering theory that can be approximated with deep networks.
- Reworked lecture part with new pre-recordings and structure
- all material available from the start of the course (but might be updated during the course)

Grading

- Assignments (2 assignments 40/60 weight): 50%
- Exam 50% (2 questions)

github.com rabbit holes

- https://github.com/alievk/avatarify
- https://github.com/CompVis/stable-diffusion
- https://github.com/deepfakes/faceswap
- https://github.com/Avik-Jain/100-Days-Of-ML-Code
- https://github.com/facebookresearch/Detectron
- https://github.com/fastai/fastai
- https://github.com/CMU-Perceptual-Computing-Lab/openpose
- https://github.com/matterport/Mask_RCNN