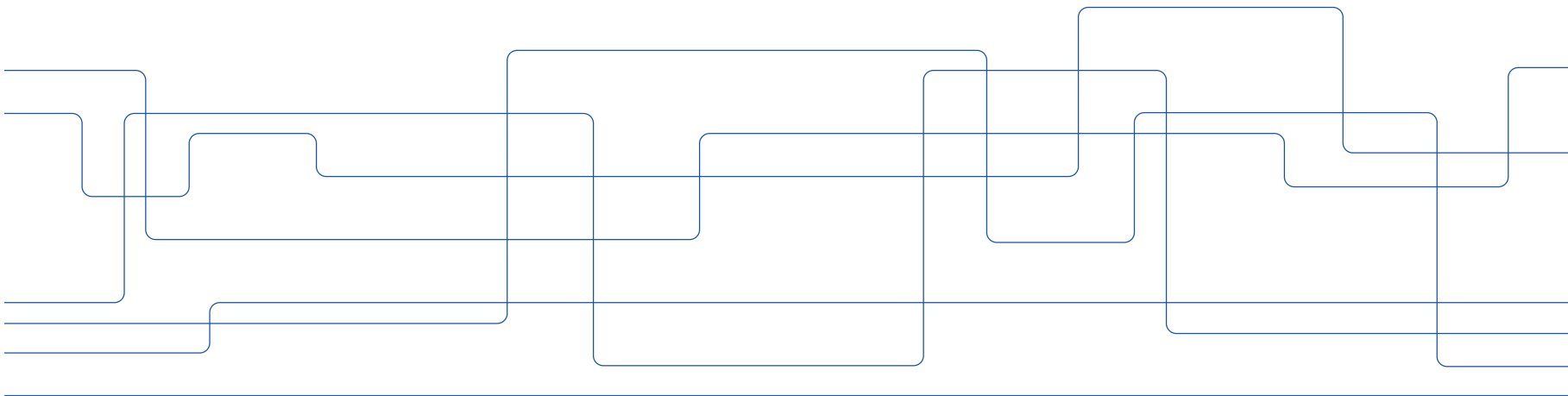




EL2805: Reinforcement Learning

Lab instructions

Author: Alessio Russo (alessior@kth.se)



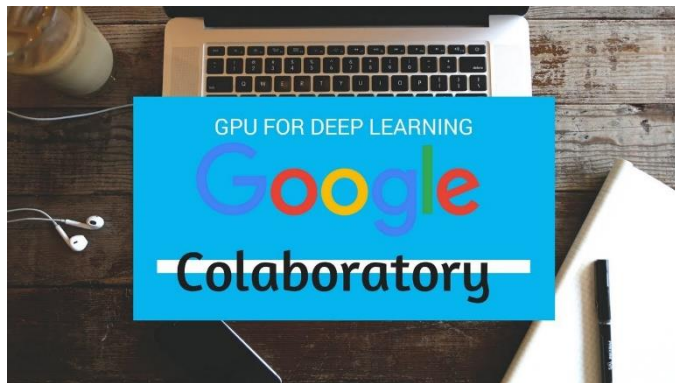


1. Prerequisites

- **We will assume that you know enough to program in Python**
 - You should know how to use a shell and run Python from it
 - You should know what is a class, an object, a variable, a function, list and dictionaries
 - Knowledge of NumPy and Matplotlib to be able to perform mathematical operations and show results
- **In case you need a refresh, you can check out the following tutorial**
<https://cs231n.github.io/python-numpy-tutorial/> (Stanford tutorial)

2. Running code on the cloud

- **If you do not want to run code on your computer, but on the cloud, you can use Google Colabs**
 - Google Colabs is a free resource from Google that allows deep learning code to be run on the cloud
 - Useful for problems that are computationally heavy



- You can use a GPU
- Most likely it is faster than your laptop
- Allows more people to collaborate together
- Most of the libraries we use are already installed
- **Link:** <https://colab.research.google.com/>

2. Running code on the cloud

Google Colab interface showing a Jupyter notebook titled "el2805_random_signal_example.ipynb". The notebook contains Python code for generating a random Gaussian signal and plotting it. The code is as follows:

```
import numpy as np
import matplotlib.pyplot as plt

N = 50
x = [i for i in range(N)]
y = np.random.randn(50)

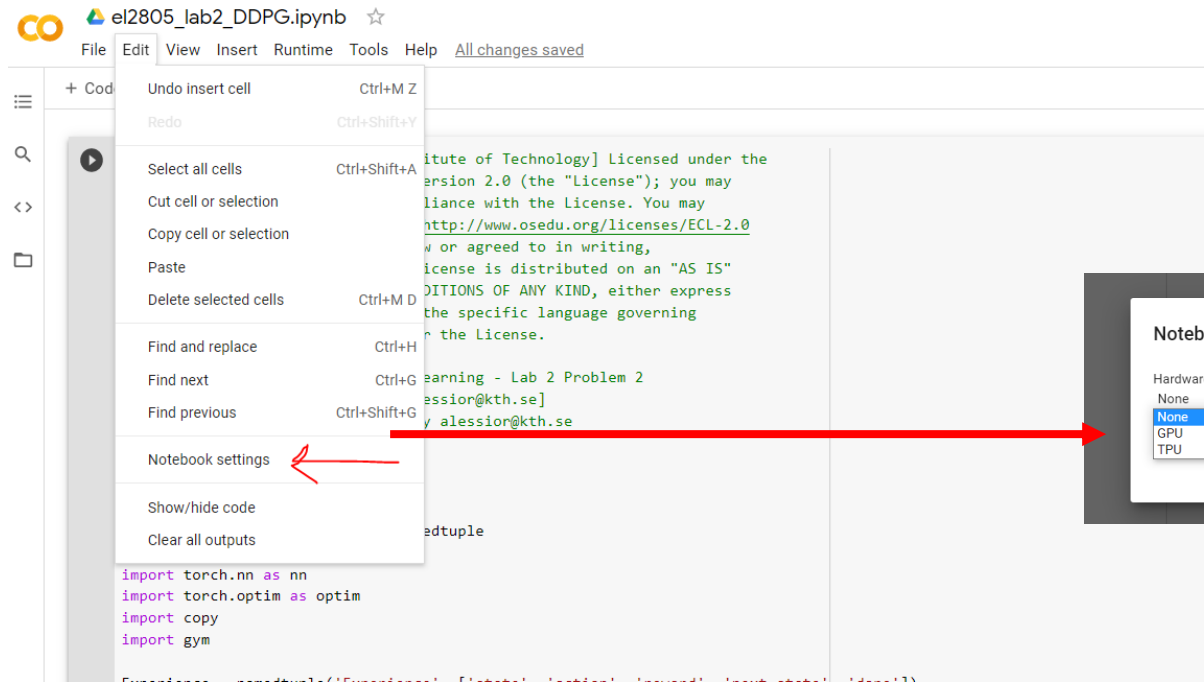
plt.plot(x, y)
plt.grid()
plt.xlabel('x')
plt.ylabel('y')
plt.title("Random Gaussian signal")
plt.show()
```

The plot shows a random Gaussian signal (y-axis) versus x (x-axis). The signal is a noisy line fluctuating around zero, with a grid overlaid. The x-axis ranges from 0 to 50, and the y-axis ranges from -2 to 2.

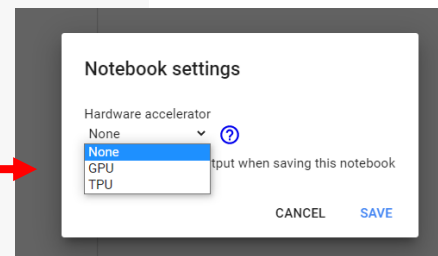
Annotations:

- It is exactly the same as Jupyter notebooks (check next slides for more info)
- Share your work
- In Google colab all the libraries are already installed. For lab2 you will have to run the command:
`!pip3 install box2d-py==2.3.8`

2. Running code on the cloud



You can also enable GPU acceleration





3. Running code on your computer: setup

- **Alternatively**, if you want to run code on your computer, you can install Python (version 3 atleast)
- **We will use Anaconda to install libraries/packages instead of Pip**
 - **Motivation:** Anaconda allows to create self-isolated virtual environments that can contain different versions of Python and different libraries. An environment is like an “island”: everything you install/put in there does not interfere with other “islands” you may have (so that it does not interfere with other environments you may have on your computer)
 - Read <https://bit.ly/2YJsgMw> for more information

3. Running code on your computer: setup

1. Download Anaconda and run the installer

1. Link: <https://www.anaconda.com/products/individual#Downloads>
2. Open the anaconda shell and run the command “`conda config --add channels conda-forge`”. This will allow Conda to fetch packages also from the conda-forge channel, which is a community effort to provide packages that are not included in the default Conda channel.

2. Create an environment for the course

1. Create a dedicated folder for the course where to save all the files.
2. Open the shell and create an environment for the course. Type “`conda create --name el2805`” to create a new environment named “el2805”

3. Other useful commands

1. Activate/deactivate the environment by typing “`conda activate el2805`” or “`conda deactivate`”
2. Install/remove packages using “`conda install`” or “`conda remove`” (for example, “`conda install numpy`”)
3. Delete an environment (and all the packages therein) by typing “`conda env remove --name el2805`”

Conda cheat-sheet →



(link: <https://bit.ly/2G4WB1y>)



3. Running code on your computer: libraries

1. Open the shell and activate the course environment (“`conda activate el2805`”)

2. Install the Basic libraries:

1. Type “`conda install numpy jupyter matplotlib scipy jupyterlab ipywidgets nodejs tqdm`” to install basic libraries and utilities
2. Type “`jupyter nbextension enable --py widgets nbextension`” to enable widgets on Jupyter notebooks
3. Type “`jupyter labextension install @jupyter-widgets/jupyterlab-manager`” to enable widgets on Jupyter lab

3. Install the Deep learning library:

1. Install the CPU version of PyTorch “`conda install pytorch torchvision cpuonly -c pytorch`”
2. You can also install the GPU version if you have CUDA installed (google “cuda install”). Check the PyTorch website <https://pytorch.org/> to install the GPU version.

4. Install the Reinforcement learning environments

1. Install OpenGym using the command “`conda install gym pybox2d`”
 1. If that does not work, run “`conda config --add channels conda-forge`” and then try again.

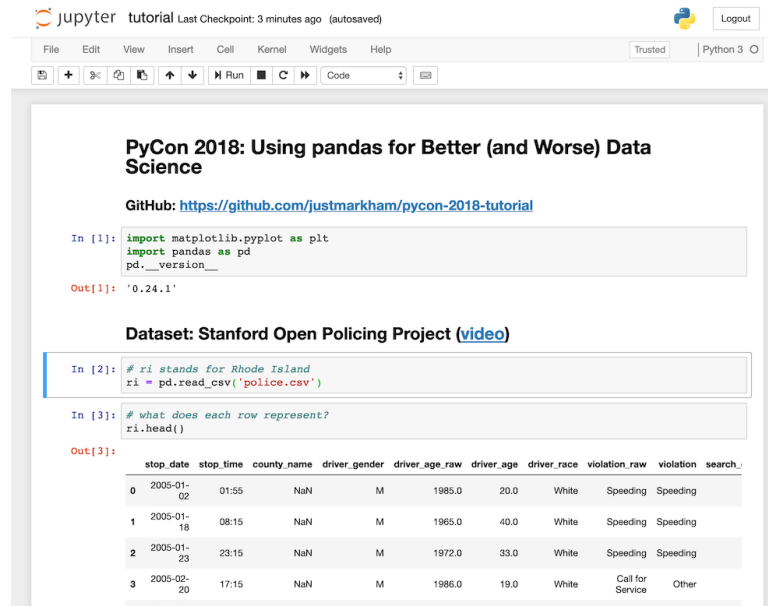
4. Running code on your computer: editors

1. If you want to use a text editor, we suggest the following ones:
 1. *Sublime Text: Nice interface, but a license must be purchased for continued use*
Link: <https://www.sublimetext.com/>
 2. *Atom: nice interface, completely free, developed by GitHub*
Link: <https://atom.io/>
2. If you do not want to use a text editor, but use a web application that runs on your computer, we suggest Jupyter Notebook or Jupyter Lab (check next slides).



5. Running code on your computer: Jupyter notebooks

- Jupyter notebooks allow one to run code interactively in a web browser.



PyCon 2018: Using pandas for Better (and Worse) Data Science

GitHub: <https://github.com/justmarkham/pycon-2018-tutorial>

```
In [1]: import matplotlib.pyplot as plt
import pandas as pd
pd.__version__
```

```
Out[1]: '0.24.1'
```

Dataset: Stanford Open Policing Project ([video](#))

```
In [2]: # ri stands for Rhode Island
ri = pd.read_csv('police.csv')
```

```
In [3]: # what does each row represent?
ri.head()
```

```
Out[3]:
```

	stop_date	stop_time	county_name	driver_gender	driver_age_raw	driver_age	driver_race	violation_raw	violation	search_
0	2005-01-02	01:55	NaN	M	1985.0	20.0	White	Speeding	Speeding	
1	2005-01-18	08:15	NaN	M	1965.0	40.0	White	Speeding	Speeding	
2	2005-01-23	23:15	NaN	M	1972.0	33.0	White	Speeding	Speeding	
3	2005-02-20	17:15	NaN	M	1986.0	19.0	White	Call for Service	Other	



5. Running code on your computer: Jupyter notebooks

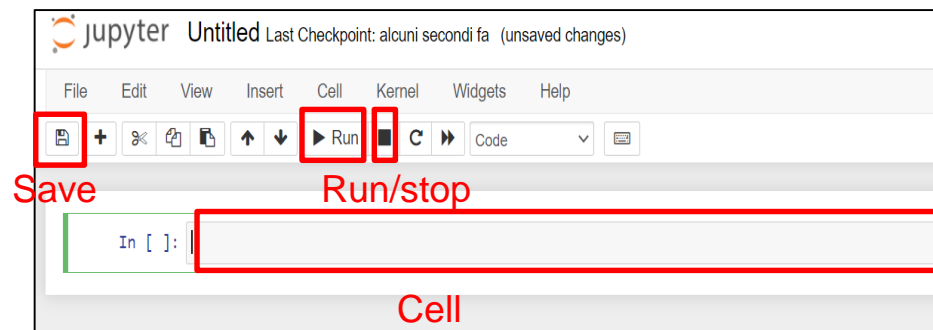
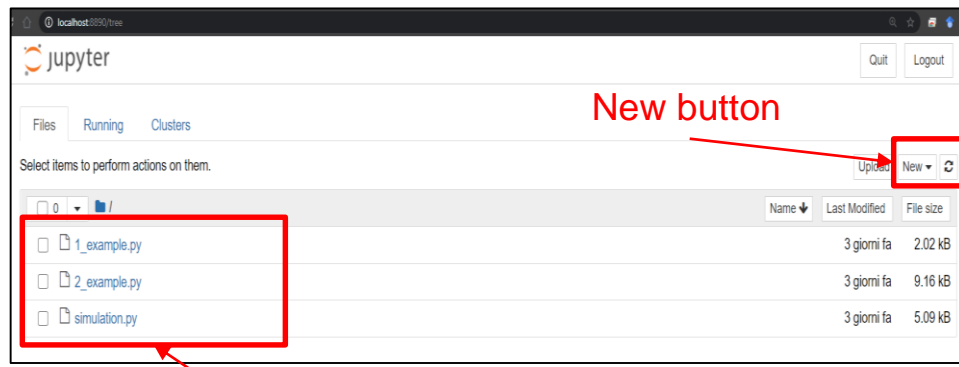
- Jupyter notebooks allow one to run code interactively in a web browser.
- **To start a notebook, open the shell and go to a folder (for example the course folder). Activate the conda course environment and then type “jupyter notebook”.**

```
Anaconda Powershell Prompt (Anaconda3)
base) PS C:\>
base) PS C:\>
base) PS C:\> conda activate el2805
el2805) PS C:\> cd 'H:\appdata\xp.V2\Documents\RL Course\'
el2805) PS H:\appdata\xp.V2\Documents\RL Course> jupyter notebook
I 19:16:55.728 NotebookApp] The port 8888 is already in use, trying another port.
I 19:16:55.730 NotebookApp] The port 8888 is already in use, trying another port.
I 19:16:56.147 NotebookApp] JupyterLab extension loaded from C:\Users\alessior\Anaconda3\envs\el2805\lib\site-packages\
jupyterlab
I 19:16:56.148 NotebookApp] JupyterLab application directory is C:\Users\alessior\Anaconda3\envs\el2805\share\jupyterlab
I 19:16:56.156 NotebookApp] Serving notebooks from local directory: H:\appdata\xp.V2\Documents\RL Course
I 19:16:56.157 NotebookApp] Jupyter Notebook 6.1.1 is running at:
I 19:16:56.157 NotebookApp] http://localhost:8890/?token=fb4e14c4e93359429788279de4c01e14e3f000327d857498
I 19:16:56.158 NotebookApp] or http://127.0.0.1:8890/?token=fb4e14c4e93359429788279de4c01e14e3f000327d857498
I 19:16:56.158 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
C 19:16:56.452 NotebookApp]

To access the notebook, open this file in a browser:
file:///C:/Users/alessior/AppData/Roaming/jupyter/runtime/nbserver-16408-open.html
Or copy and paste one of these URLs:
http://localhost:8890/?token=fb4e14c4e93359429788279de4c01e14e3f000327d857498
or http://127.0.0.1:8890/?token=fb4e14c4e93359429788279de4c01e14e3f000327d857498
```

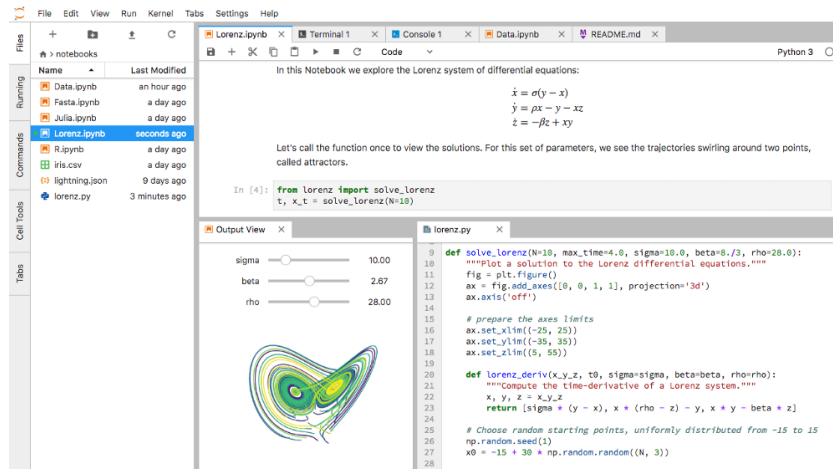
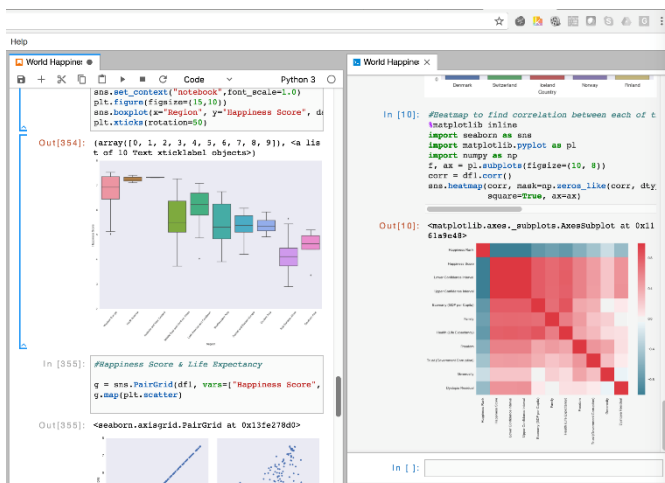
5. Running code on your computer: Jupyter notebooks

- Jupyter notebooks allow one to run code interactively in a web browser.
- To start a notebook, open the shell and go to a folder (for example the course folder). Activate the conda course environment and then type “**jupyter notebook**”.
- This will open a new tab in your browser, showing the list of files in the folder. You can open a file or create a new one.
- If you want to create a new file, click on the button “New”. This will create a file named “Untitled.ipynb” inside the chosen folder.
- You can execute code inside the cells.



6. Running code on your computer: Jupyter labs

- Jupyter lab is an interactive interface where you can open several files and notebooks as tabs in the same window, like an IDE.



6. Running code on your computer: Jupyter labs

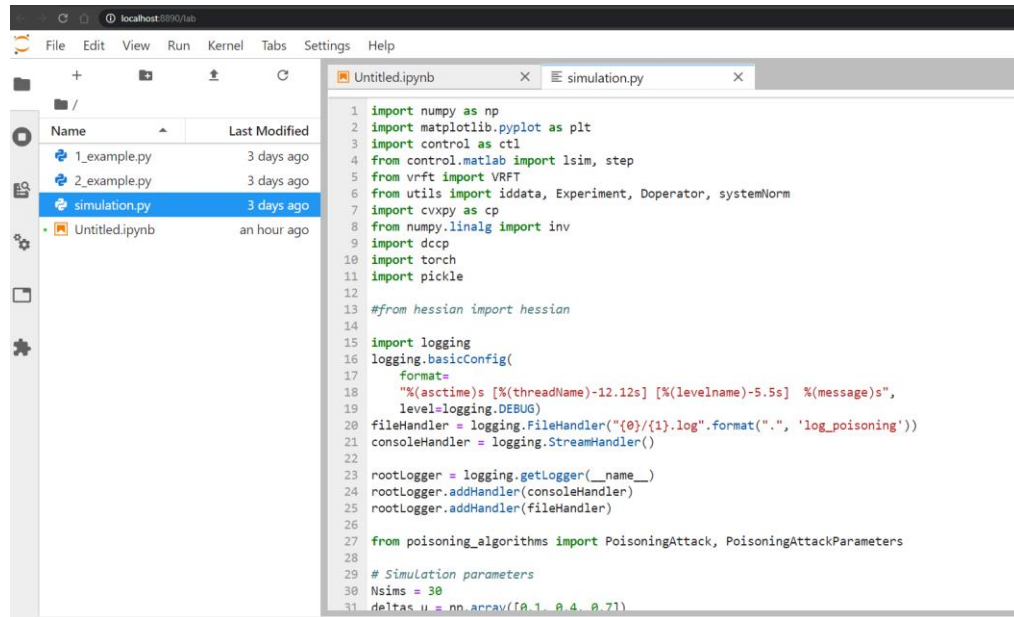
- To start a jupyter lab, open the shell and go to a folder (for example the course folder). Activate the conda course environment and then type “jupyter lab”.

```
(base) PS C:\> conda activate el2805
(el2805) PS C:\> cd 'H:\appdata\xp.V2\Documents\RL Course\'
(el2805) PS H:\appdata\xp.V2\Documents\RL Course> jupyter lab
[I 20:15:35.068 LabApp] The port 8888 is already in use, trying another port.
[I 20:15:35.070 LabApp] The port 8889 is already in use, trying another port.
[I 20:15:35.135 LabApp] JupyterLab extension loaded from C:\Users\aleessor\Anaconda3\envs\el2805\lib\site-packages\jupyterlab
[I 20:15:35.135 LabApp] JupyterLab application directory is C:\Users\aleessor\Anaconda3\envs\el2805\share\jupyterlab
[I 20:15:35.140 LabApp] Serving notebooks from local directory: H:\appdata\xp.V2\Documents\RL Course
[I 20:15:35.141 LabApp] Jupyter Notebook 6.1.1 is running at:
[I 20:15:35.141 LabApp] http://localhost:8890/?token=e7f74df3e62ff820d61bb02d88013ff16248c6e8a58179df
[I 20:15:35.141 LabApp] or http://127.0.0.1:8890/?token=e7f74df3e62ff820d61bb02d88013ff16248c6e8a58179df
[I 20:15:35.142 LabApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 20:15:35.359 LabApp]

To access the notebook, open this file in a browser:
file:///C:/Users/aleessor/AppData/Roaming/jupyter/runtime/nbserver-8388-open.html
Or copy and paste one of these URLs:
http://localhost:8890/?token=e7f74df3e62ff820d61bb02d88013ff16248c6e8a58179df
or http://127.0.0.1:8890/?token=e7f74df3e62ff820d61bb02d88013ff16248c6e8a58179df
[W 20:15:39.381 LabApp] Could not determine jupyterlab build status without nodejs
[W 20:15:40.975 LabApp] 404 GET /api/contents/environment.yml?content=0&1598811340923 (::1): No such file or directory:
environment.yml
[W 20:15:40.977 LabApp] No such file or directory: environment.yml
[W 20:15:40.979 LabApp] 404 GET /api/contents/environment.yml?content=0&1598811340923 (::1) 27.99ms referer=http://localhost:8890/lab
[I 20:15:42.708 LabApp] Kernel started: ea2302bc-234d-4eb0-9c45-799cadedf134f, name: python3
```

6. Running code on your computer: Jupyter labs

- To start a jupyter lab, open the shell and go to a folder (for example the course folder). Activate the conda course environment and then type “**jupyter lab**”.
- **This will open a new tab on your browser where you can select files, create new ones and work in an integrated interface**



The screenshot shows the Jupyter Lab interface running on localhost:8890. On the left, a file browser displays a directory structure with files: 1_example.py, 2_example.py, simulation.py, and Untitled.ipynb. The 'simulation.py' file is selected. On the right, the code editor shows the contents of 'simulation.py', which includes imports for numpy, matplotlib, control, vrft, utils, cvxpy, torch, and pickle, followed by logging configuration and simulation parameters.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import control as ctl
4 from control.matlab import lsim, step
5 from vrft import VRFT
6 from utils import iddata, Experiment, Doperator, systemNorm
7 import cvxpy as cp
8 from numpy.linalg import inv
9 import dccp
10 import torch
11 import pickle
12
13 #from hessian import hessian
14
15 import logging
16 logging.basicConfig(
17     format=
18         "%(asctime)s [%(threadName)-12.12s] [%(levelname)-5.5s] %(message)s",
19         level=logging.DEBUG)
20 fileHandler = logging.FileHandler("{0}/{1}.log".format(".", 'log_poisoning'))
21 consoleHandler = logging.StreamHandler()
22
23 rootLogger = logging.getLogger(__name__)
24 rootLogger.addHandler(consoleHandler)
25 rootLogger.addHandler(fileHandler)
26
27 from poisoning_algorithms import PoisoningAttack, PoisoningAttackParameters
28
29 # Simulation parameters
30 Nsims = 30
31 deltas_u = np.array([0.1, 0.4, 0.7])
```