



The Ottawa
Hospital

RESEARCH
INSTITUTE

L'Hôpital
d'Ottawa

INSTITUT DE
RECHERCHE



uOttawa

Institut de recherche
sur le cerveau

Brain and Mind
Research Institute

Workshop on Applied Deep Learning in Intracranial Neurophysiology

Part 1 – Getting Started with Deep Learning Tools

September 16, 2019

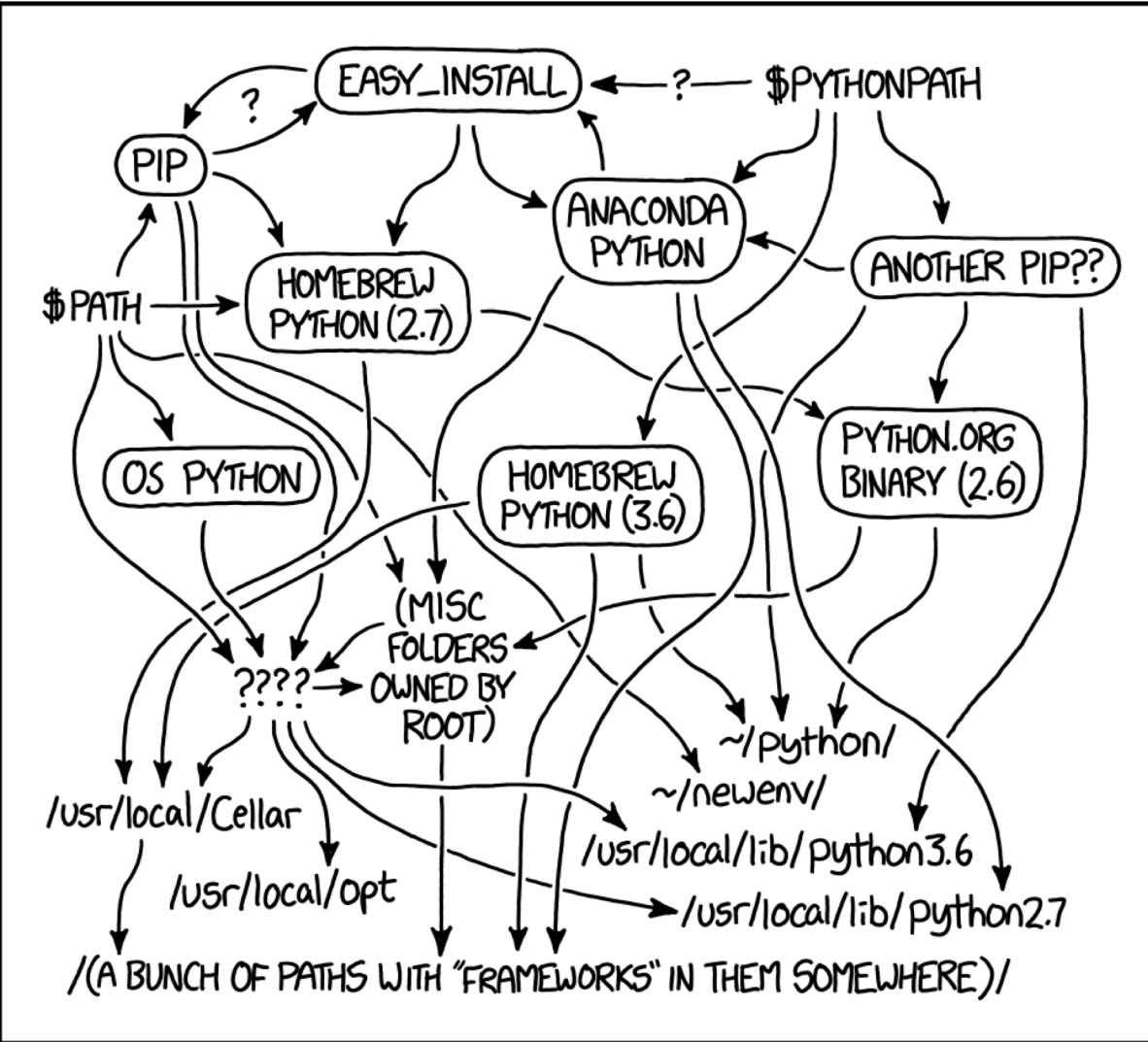
Presented by Chadwick Boulay, MSc, PhD

Ottawa Hospital Research Institute

University of Ottawa

Outline

- Managing Python environments
- Working in your Python environment
- Jupyter notebooks
- Basic data exploration



MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED
THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.



Leading Open Data Science Platform Powered by Python



Leading Package and Environment Manager

OPEN DATA SCIENCE



DATA



COMPUTATION



RESULTS

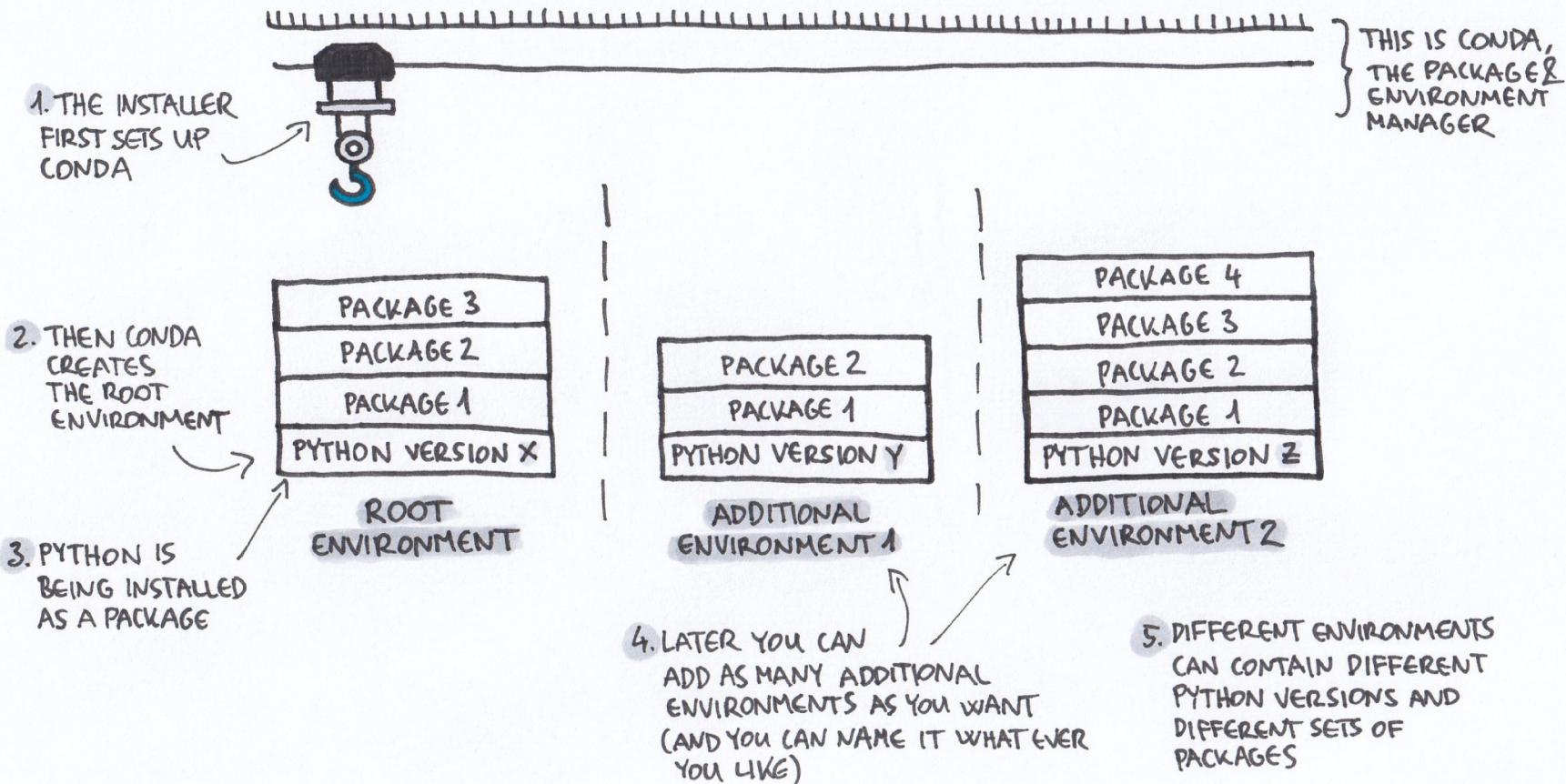
- ✓ Reproducibility for packages and environments
- ✓ Building interactive visualizations

[Get miniconda](#)

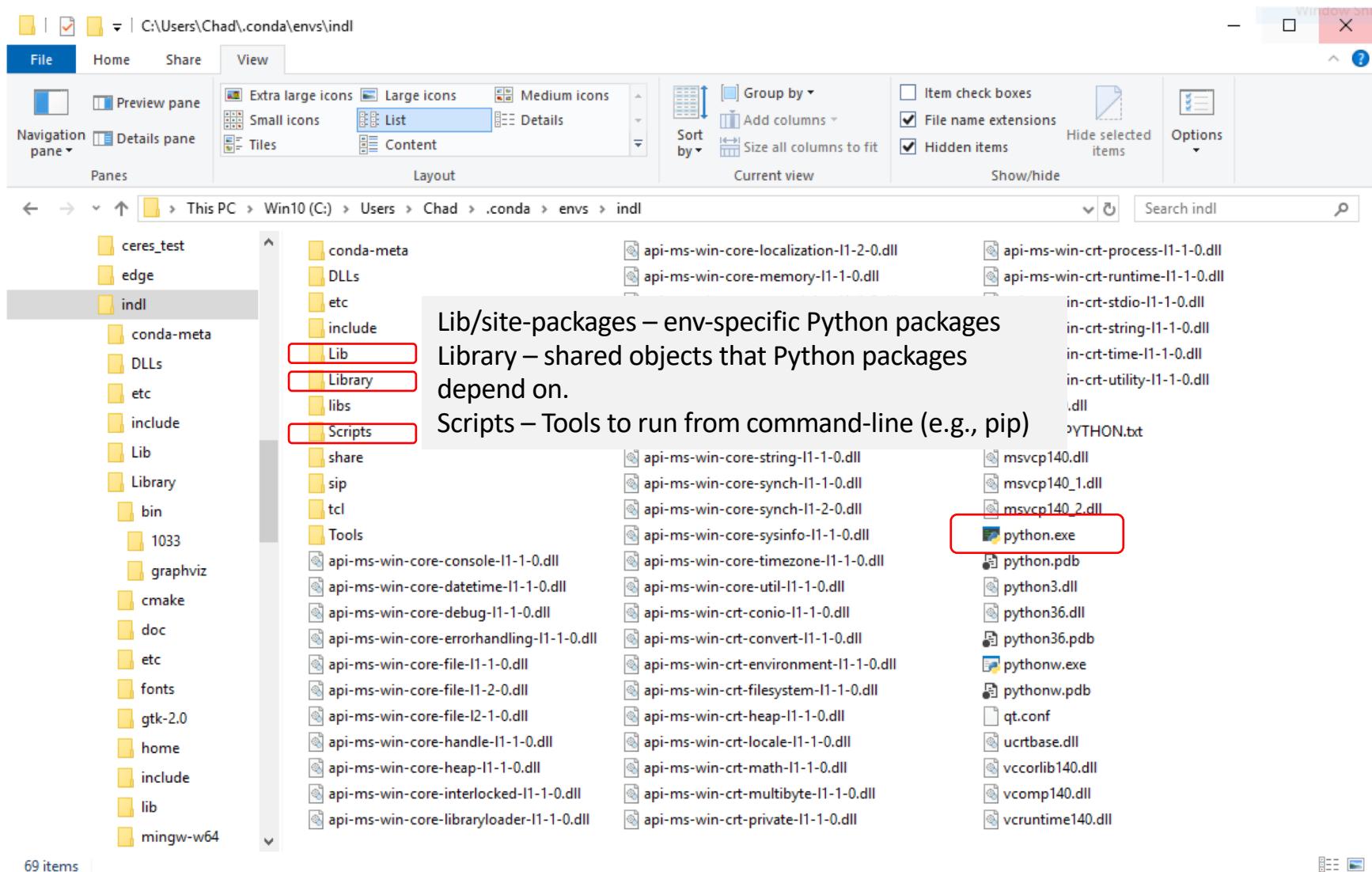
conda is a package and environment manager. How does that work?

- After installing miniconda, open an “Anaconda Prompt” or Terminal
- Create an environment
 - `$conda create --name indl python=3.6`
 - creates a folder somewhere (`~/.conda`) with everything Python needs all laid out in a known folder structure (see next slides).
- Activate an environment
 - Windows: `$conda activate indl`
 - Mac/Linux: `$source activate indl`
 - Sets the PATH (see later slides)
- Maintain packages for the active environment
 - install, update, handle version dependencies
 - (indl) `$conda install <list of packages>`
- Some packages not available through conda
 - Use alternate channels: (np) `$conda install <pkg> -c conda-forge`
 - Use pip: (np) `$pip install <pkg>`
- [More info](#)

Anatomy of a Python environment



Anatomy of a Python environment



(Windows) What does it mean to “activate an environment”?

- Binaries (.exe, .dll) that are linked to other Dynamic Link Libraries (DLLs) need to know where to find them at run-time. They use a search strategy.
 - Full answer is complicated. Simple version:
binary dir > C:\Windows\System > C:\Windows\SysWOW64 > C:\Windows > cwd > PATH dirs.
- echo %PATH%
 - C:\Program Files (x86)\Microsoft Visual Studio 14.0\Common7\IDE\CommonExtensions\Microsoft\TestWindow;C:\Program Files (x86)\MSBuild\14.0\bin\amd64;C:\Program Files (x86)\Microsoft Visual Studio 14.0\VC\BIN\amd64;C:\Windows\Microsoft.NET\Framework64\v4.0.30319;C:\Program Files (x86)\Microsoft Visual Studio 14.0\VC\VPackages;C:\Program Files (x86)\Microsoft Visual Studio 14.0\Common7\IDE;C:\Program Files (x86)\Microsoft Visual Studio 14.0\Common7\Tools;C:\Program Files (x86)\HTML Help Workshop;C:\Program Files (x86)\Microsoft Visual Studio 14.0\Team Tools\Performance Tools\x64;C:\Program Files (x86)\Microsoft Visual Studio 14.0\Team Tools\Performance Tools;C:\Program Files (x86)\Windows Kits\10\bin\x64;C:\Program Files (x86)\Windows Kits\10\bin\x86;C:\Program Files (x86)\Microsoft SDKs\Windows\v10.0A\bin\NETFX 4.6.1\Tools\x64;C:\Users\Chad\conda\envs\indl;C:\Users\Chad\conda\envs\indl\Library\mingw-w64\bin;C:\Users\Chad\conda\envs\indl\Library\usr\bin;C:\Users\Chad\conda\envs\indl\Library\bin;C:\Users\Chad\conda\envs\indl\Scripts;C:\Users\Chad\conda\envs\indl\bin;C:\ProgramData\Miniconda3\condabin;C:\Program Files (x86)\Measurement Computing\DAQ;C:\Windows\System32;C:\Windows\System32\WBem;C:\Windows\System32\WindowsPowerShell\v1.0;C:\Windows\System32\OpenSSH;C:\Program Files\Git\cmd;C:\Program Files (x86)\MATLAB\MATLAB Compiler Runtime\v716\runtime\win32;C:\Users\Chad\dnx\bin;C:\Program Files\Microsoft DNvm;C:\Program Files\Microsoft SQL Server\130\Tools\Binn;C:\WINDOWS\System32;C:\WINDOWS\System32\WBem;C:\WINDOWS\System32\WindowsPowerShell\v1.0;C:\WINDOWS\System32\OpenSSH;C:\Program Files (x86)\NVIDIA Corporation\PhysX\Common;C:\Program Files\CMake\bin;C:\Users\Chad\AppData\Local\Microsoft\WindowsApps;;C:\Program Files\Mercurial;C:\ProgramData\Miniconda3\lib\site-packages\pywin32_system32;C:\ProgramData\Miniconda3\lib\site-packages\pywin32_system32;C:\Program Files (x86)\Windows Kits\10\bin\10.0.17763.0\x86
- Debugging missing libraries (DLLNotFound)
 - From the Anaconda Prompt, run Dependencies.exe, and open the binary object that is failing to load.
 - Identify any missing libraries. Download and install them if you truly do not have them. If you have them but they are not being found, add their directory to the PATH.

cwd: current working directory

(Mac/Linux) What does it mean to “activate an environment”?

- `$echo $PATH`

```
/Volumes/STORE/Tools/Misc/miniconda3/envs/indl/bin:/usr/local/opt/ruby/bin:/usr/local/opt/ruby/bin:/Volumes/STORE/Tools/Misc/miniconda3/bin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/Library/Frameworks/Mono.framework/Versions/Current/Commands
```

- Still has the conda environment path
 - Many fewer paths than Windows
 - Only used to find executables, not link paths.
- Search Strategy:
 - `~/lib`, `/usr/local/lib`, and `/usr/lib`
 - `LD_LIBRARY_PATH`
 - A list of paths to search for dynamic libraries
 - `RPATH`
 - Each file, for each dependency, can have a list of search paths.
 - [Debugging missing libraries in Linux](#)

Recommended packages for Data Science

Name	Notes
numpy	Linear algebra and array (matrix) ops
scipy	Signal processing, some data i/o tools
pandas	DataFrame, organize data in tables
scikit-learn	Machine learning (not DL)
ipython	Prettier interpreter
matplotlib	Standard plotting library
plotly	Better plotting
h5py	Data container format for file i/o
statsmodels	
python-neo	Imports many neurophys data formats

Matlab vs Python: Quick Tips

Matlab

- 1-based indexing
- Access 3rd party functions by “add to path”
- White space doesn’t matter
 - Flow control blocks require `end`
- eol: ;

Numpy

- 0-based indexing
- Access 3rd party functions by `import`
 - import numpy as np
- White space very important
 - End block with de-indent (4 spaces!)
- eol: <newline>
 - \ to cont. on next line

Matlab vs Python (Numpy): Matrix Memory Layout

Matlab

- Fortran order
- Column-major

	24	28	32	
	12	16	20	3
0	4	8	21	2
1	5	9	22	4
2	6	10	23	5
3	7	11		

- First index is fastest
 - `my_mat(r, c)`

Numpy

- C order (default)
- Row-major

	24	25	26	
	12	13	14	29
0	1	2	17	32
3	4	5	20	35
6	7	8	23	
9	10	11		

- Last index is fastest
 - `my_mat[r, c] == my_mat[r][c]`
 - Arrays of arrays of arrays

Deep Learning in Practice: Local vs Remote Configurations

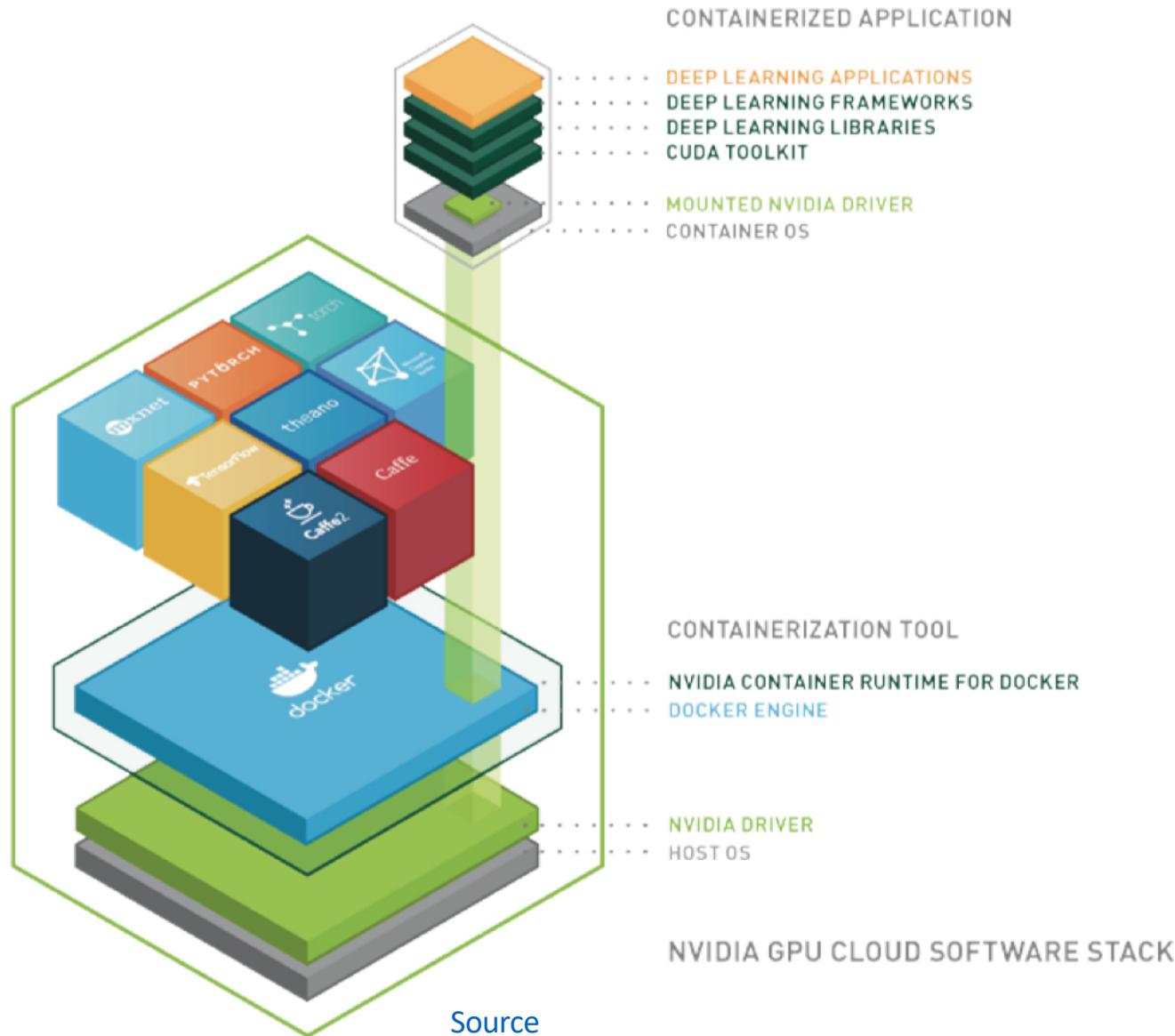
Local

- + Infinitely flexible
- + Easier to deal with large single-source data
(common in neuroscience)
- DL requires expensive and bulky hardware (GPU)
- Requires configuration knowledge

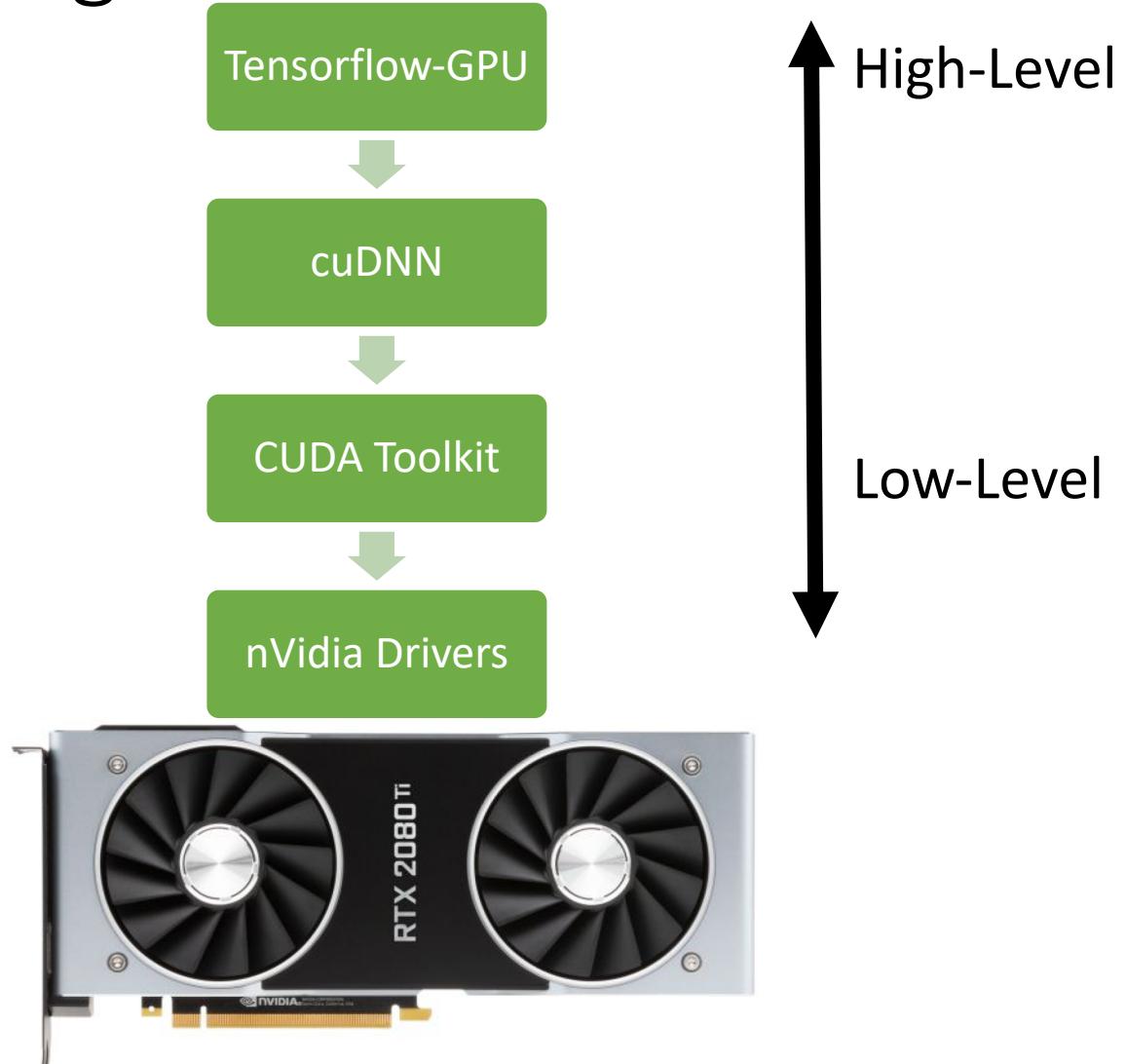
Remote

- + Can be free
- + Configuration handled for you
- Free services come with constraints
- Managing data can be tricky
- Paid services can get expensive

Remote Configurations



Local Configurations



Recommended packages for Deep Learning (Local Configurations Only)

Required

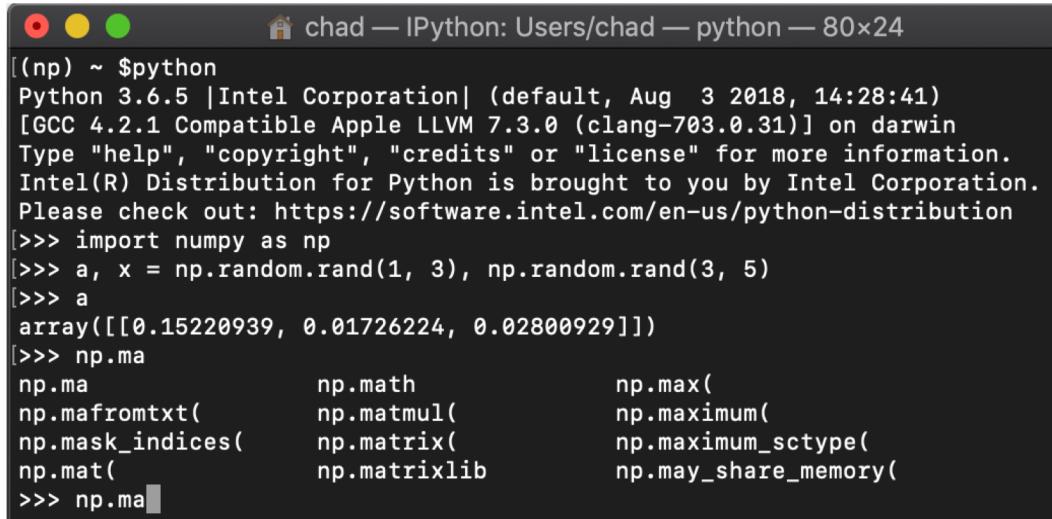
- tensorflow-gpu – Deep Learning framework from google, GPU-enabled variant
 - Each version of [tensorflow requires specific versions of cuDNN and cudatoolkit](#), each version of [cudatoolkit requires specific versions of nVidia graphics drivers](#).
 - Make sure cuDNN, CUDA, and CUPTI are all on the PATH!
- jax – in progress

Optional

- pytorch – Deep Learning framework from Facebook
- fastai – High-level API around pytorch, with many customizations; accompanies a popular online DL course.

Working in Python

- Open Anaconda prompt/terminal
- `$python myscript.py`
- `$python`

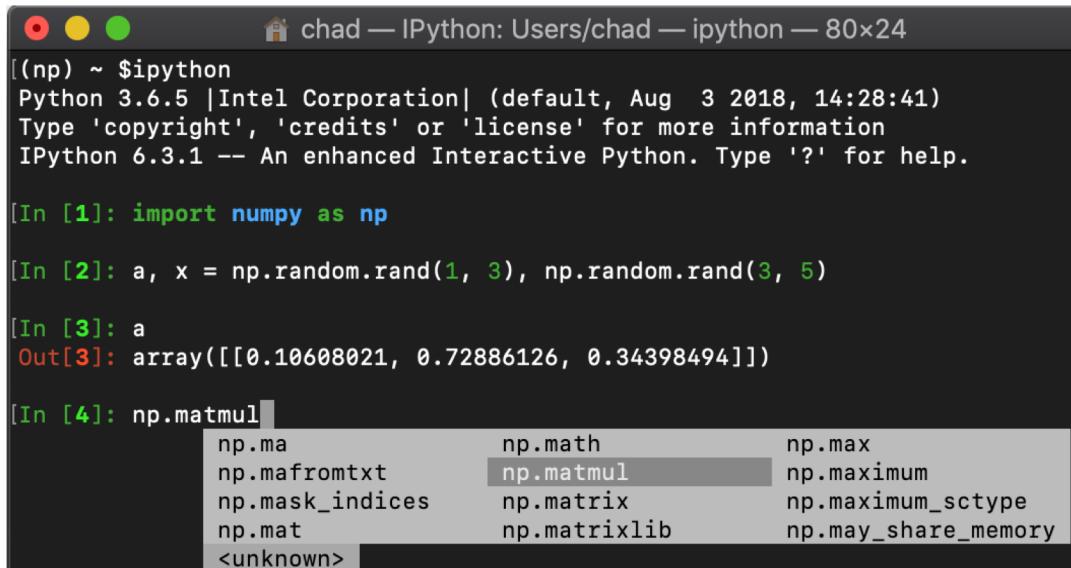


```
[np] ~ $python
Python 3.6.5 |Intel Corporation| (default, Aug  3 2018, 14:28:41)
[GCC 4.2.1 Compatible Apple LLVM 7.3.0 (clang-703.0.31)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
Intel(R) Distribution for Python is brought to you by Intel Corporation.
Please check out: https://software.intel.com/en-us/python-distribution
[>>> import numpy as np
[>>> a, x = np.random.rand(1, 3), np.random.rand(3, 5)
[>>> a
array([[0.15220939, 0.01726224, 0.02800929]])
[>>> np.ma
np.ma          np.math          np.max(
np.mafromtxt()  np.matmul()      np.maximum(
np.mask_indices() np.matrix()    np.maximum_sctype(
np.mat()        np.matrixlib   np.may_share_memory(
[>>> np.ma]
```

- `$ipython`

Prettier

Better tab-completion



```
[np] ~ $ipython
Python 3.6.5 |Intel Corporation| (default, Aug  3 2018, 14:28:41)
Type 'copyright', 'credits' or 'license' for more information
IPython 6.3.1 -- An enhanced Interactive Python. Type '?' for help.

[In 1]: import numpy as np
[In 2]: a, x = np.random.rand(1, 3), np.random.rand(3, 5)

[In 3]: a
Out[3]: array([[0.10608021, 0.72886126, 0.34398494]])

[In 4]: np.matmul
np.ma          np.math          np.max(
np.mafromtxt()  np.matmul()      np.maximum(
np.mask_indices() np.matrix()    np.maximum_sctype(
np.mat()        np.matrixlib   np.may_share_memory(
<unknown> ]
```

Working in Python

- \$jupyter-notebook

```
(*) IntracranialNeurophysDL — IPython: Users/chad — jupyter-notebook — 119x24
[In 1]: In[1]: IntracranialNeurophysDL $jupyter-notebook
[I 11:11:22.737 NotebookApp] Serving notebooks from local directory: /Volumes/STORE/Tools/Neurophys/IntracranialNeurophysDL
[I 11:11:22.737 NotebookApp] 0 active kernels
[I 11:11:22.737 NotebookApp] The Jupyter Notebook is running at:
[I 11:11:22.737 NotebookApp] http://localhost:8888/?token=a1654f2d0b6736c9d04ce1d4f69022e4943ff57bef5cb370
[I 11:11:22.737 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 11:11:22.740 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
http://localhost:8888/?token=a1654f2d0b6736c9d04ce1d4f69022e4943ff57bef5cb370
[I 11:11:22.917 NotebookApp] Accepting one-time-token-authenticated connection from ::1
```

The screenshot shows a Jupyter Notebook interface with the title "jupyter 01_01_data_import (unsaved changes)". The toolbar includes File, Edit, View, Insert, Cell, Kernel, Help, Run, Code, and a Python 3 kernel selector. The status bar indicates "Not Trusted".

In [28]:

```
import pandas as pd

# Get the 'signals' chunk
chunk_names = [_[0] for _ in chunks]
chunk = chunks[chunk_names.index('signals')][1]
ax_types = [_[‘type’] for _ in chunk[‘axes’]]

print("The 'signals' chunk has data with shape {}".format(chunk[‘data’].shape))
print("The axes types are {}".format(ax_types))

instance_axis = chunk[‘axes’][ax_types.index(‘instance’)]
print("The trial label frequencies are \n{}".format(pd.value_counts(instance_axis[‘data’])[‘Marker’]))
```

The data contain 603 trials, with each trial having 17 samples in time, and 31 channels. Of the 603 trials, 303 or inter-stimulus intervals, and the remaining 300 are split between face and house.

When a data container has more than 2 axes (or dimensions), it is no longer a "matrix", it is a "tensor". Let's use a tensor-decomposition tool to get a simpler view of the tensor contents.

Working in Python

- IDE (PyCharm, VS Code, Spyder)

Using PyCharm, open the repo as a “project”

The screenshot shows the PyCharm IDE interface. On the left is the Project tool window, which lists several files and folders under the project 'IntracranialNeurophysDL'. In the center is the code editor showing a file named '03_convert.py'. The code defines variables like TVLDA_SEGMENT, PSD_SEGMENT, etc., and contains a main block with file operations. A red arrow points from a callout box to the 'Configure Python interpreter' button in the top right corner of the code editor. The callout box contains the text: 'Configure interpreter Click on this tip Or use the Preferences/Settings menu.' At the bottom right of the code editor, there is another callout box with the text: 'Looks like you're using NumPy Would you like to turn scientific mode on? Use scientific mode Keep current layout...'.

```
# -*- coding: utf-8 -*-
Created on Sun Jan 20 21:07:28 2019
@author: Chadwick Boulay
@author: Anahita Malvea
This must be run from the ../../ directory (parent/parent)
"""

import ...

TVLDA_SEGMENT = [-0.2, 0.6]
PSD_SEGMENT = [-0.3, 0.7]
ERP_SEGMENT = [-0.2, 0.4]
ERP_BASELINE = [-0.2, 0.05]
KEEP_F_BANDS = [1.5, 4,
                 4, 7,
                 8, 15,
                 16, 31,
                 32, 55,
                 70, 130,
                 130, 350]
WIN_DUR = 0.05
NTRIALS = 300

if __name__ == "__main__":
    logging.basicConfig(level=logging.DEBUG)

    working_dir = Path.cwd() / 'data' / 'kjm_ecog'
    studies_file = working_dir / 'studies.csv'

    # Get the list of studies
    studies = []
    with open(studies_file) as csvfile:
        datasetreader = csv.DictReader(csvfile, delimiter=',', quotechar='''')
        for row in datasetreader:
            studies.append(row)

    datasets_file = working_dir / 'datasets.csv'
    datasets = []
    with open(datasets_file) as csvfile:
        datasetreader = csv.DictReader(csvfile, delimiter=',', quotechar='''')
        for row in datasetreader:
```

Preferences

Project: IntracranialNeurophys... > Project Interpreter

For current project

Reset

▼ Appearance & Behavior

Appearance

Menus and Toolbars

► System Settings

File Colors

Scopes

Notifications

Quick Lists

Keymap

► Editor

Plugins

► Version Control

▼ Project: IntracranialNeurophys... □

Project Interpreter

Project Structure

► Build, Execution, Deployment

► Languages & Frameworks

► Tools



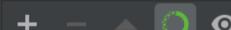
Project Interpreter: Python 3.5 (np) /Volumes/STORE/Tools/Misc/miniconda3/envs/np/bin/python

2



Add...

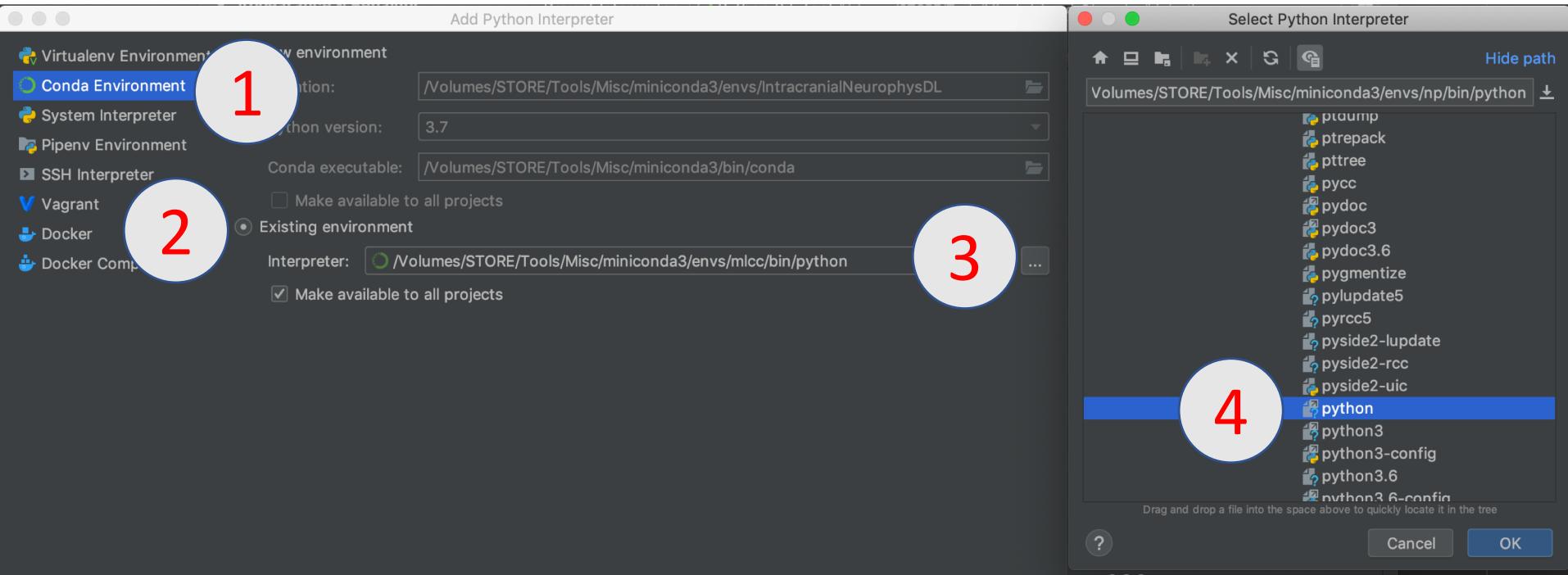
Package	Version	Latest version
aniso8601	3.0.2	▲ 6.0.0
appnope	0.1.0	0.1.0
argh	0.26.2	0.26.2
asn1crypto	0.24.0	0.24.0
backcall	0.1.0	0.1.0
backports	1.0	1.0
bleach	2.1.3	▲ 3.1.0
blist	1.3.6	1.3.6
bzip2	1.0.6	1.0.6
certifi	2018.1.18	▲ 2019.3.9
cffi	1.11.5	▲ 1.12.3
chardet	3.0.4	3.0.4
click	6.7	▲ 7.0
cryptography	2.3	▲ 2.6.1
cycler	0.10.0	0.10.0
cython	0.28.5	▲ 0.29.7
daal	2019.1	▲ 2019.3
daal4py	0.2019.1	▲ 2019.3
decorator	4.3.0	▲ 4.4.0
entrypoints	0.2.3	▲ 0.3
flask	0.12.2	▲ 1.0.3
flask-restful	0.3.6	0.3.6
freetype	2.9	▲ 2.10.0
get_terminal_size	1.0.0	1.0.0
h5py	2.9.0	▲ 2.9.0



Cancel

Apply

OK



1. Choose “Conda Environment”
2. Choose “Existing environment”
3. Click on “...”
4. Navigate to your Python interpreter (python.exe)
 - Typically in C:/Users/<name>/.conda/envs/<env name>/bin/

- Debug
`<>/ndl/cnn_helper_1D.py`
- Demo
 - Breakpoints
 - Variable inspector
 - Console
 - Step
 - Call Stack
 - Run/Debug Configuration

File 00_00_debug_and_profile.py

```

1 import numpy as np
2
3 def generate_uneven_n_fft(X, S):
4     y = (np.abs(S) * X).sum(1)
5     if uneven_n_fft:
6         y = y[::2]
7     return y
8
9 if __name__ == '__main__':
10    import time
11    FS = 30000
12    DURATION = 10
13    NCHANNELS = 10
14    t = np.linspace(0, DURATION, FS * DURATION)
15    signal = np.sin(2 * np.pi * 10 * t)
16    pink_noise = np.random.pinknoise(t, NCHANNELS)
17    signal += pink_noise
18    signal *= 0.1
19    fig, axes = plt.subplots(nrows=2, ncols=NCHANNELS)
20    axes[0, 0].plot(t, signal)
21    axes[1, 0].plot(t, signal)
22    # plot
23    axes[0, 0].set_title('Raw Signal')
24    axes[1, 0].set_title('Pink Noise')
25    axes[0, 0].grid(True)
26    axes[1, 0].grid(True)
27    axes[0, 0].set_xlim(0, 10)
28    axes[1, 0].set_xlim(0, 10)
29    axes[0, 0].set_xlabel('Time (s)')
30    axes[1, 0].set_xlabel('Time (s)')
31    axes[0, 0].set_ylabel('Amplitude')
32    axes[1, 0].set_ylabel('Amplitude')
33    axes[0, 0].set_xticks([0, 2, 4, 6, 8, 10])
34    axes[1, 0].set_xticks([0, 2, 4, 6, 8, 10])
35    axes[0, 0].set_yticks([-1, 0, 1])
36    axes[1, 0].set_yticks([-1, 0, 1])
37    fig.tight_layout()
38    plt.show()
39
40

```

Project 00_00_debug_and_profile.py

File 00_00_debug_and_profile.py

Context menu options:

- Close
- Close Others
- Close All
- Close All but Pinned
- Copy Path Ctrl+Shift+C
- Copy Relative Path Ctrl+Alt+Shift+C
- Split Vertically
- Split Horizontally
- Pin Tab
- Tabs Placement
- Sort Tabs By Filename
- Open New Tabs At The End
- Select Next Tab Alt+Right
- Select Previous Tab Alt+Left
- Reopen Closed Tab
- Add to Favorites
- Add All To Favorites
- Rename File...
- Run '00_00_debug_and_profile.py' Ctrl+Shift+F10
- Debug '00_00_debug_and_profile.py'
- Run '00_00_debug_and_profile.py' with Coverage
- Profile '00_00_debug_and_profile.py'
- Concurrency Diagram for '00_00_debug_and_profile.py' (7, 7)
- Create '00_00_debug_and_profile.py'...
- Show in Explorer
- Open in Terminal
- Local History
- Git
- Create Gist...

Jupyter Notebooks

- Demo 01_02_data_import.ipynb
- Things to try:
 - Run from console & browser
 - [Run on Google Colab](#)
- Break time and we will help you with any issues.