

Jupyter extensions

Installation

- The list of commands to install extensions with pip

```
>>>pip install jupyter_contrib_nbextensions
```

```
>>>jupyter contrib nbextension install --user
```

```
>>>jupyter nbextensions_configurator enable --user
```

- Link to forum <https://forums.fast.ai/t/useful-jupyter-notebook-tips-plugins-collapsible-sections/17919>

Configurable nbextensions

☐ disable configuration for nbextensions without explicit compatibility (they may break your

filter:

by description, section, or tags

☐ (some) LaTeX environments for Jupyter

☐ AutoSaveTime

☐ Code prettify

☒ Collapsible Headings

☐ Equation Auto Numbering

☐ Exercise2

☐ 2to3 Converter

☐ Autoscroll

☐ Codefolding

☐ Comment/Uncomment Hotkey

☒ ExecuteTime

☐ Export Embedded HTML

Some useful extensions

Configurable nbextensions

☐ disable configuration for nbextensions without explicit compatibility (they may break your notebook environment, but can be useful to show for nbextension development)

filter: by description, section, or tags

- | | | | |
|--|---|--|---|
| <input type="checkbox"/> (some) LaTeX environments for Jupyter | <input type="checkbox"/> 2to3 Converter | <input type="checkbox"/> AddBefore | <input type="checkbox"/> Autopep8 |
| <input type="checkbox"/> AutoSaveTime | <input type="checkbox"/> Autoscroll | <input type="checkbox"/> Cell Filter | <input type="checkbox"/> Code Font Size |
| <input type="checkbox"/> Code prettify | <input type="checkbox"/> Codefolding | <input type="checkbox"/> Codefolding in Editor | <input type="checkbox"/> CodeMirror mode extensions |
| <input checked="" type="checkbox"/> Collapsible Headings | <input type="checkbox"/> Comment/Uncomment Hotkey | <input checked="" type="checkbox"/> contrib_nbextensions_help_item | <input type="checkbox"/> datestamper |
| <input type="checkbox"/> Equation Auto Numbering | <input checked="" type="checkbox"/> ExecuteTime | <input type="checkbox"/> Execution Dependencies | <input type="checkbox"/> Exercise |
| <input type="checkbox"/> Exercise2 | <input type="checkbox"/> Export Embedded HTML | <input type="checkbox"/> Freeze | <input type="checkbox"/> Gist-it |
| <input type="checkbox"/> Help panel | <input type="checkbox"/> Hide Header | <input type="checkbox"/> Hide input | <input type="checkbox"/> Hide input all |
| <input type="checkbox"/> Highlight selected word | <input type="checkbox"/> highlighter | <input checked="" type="checkbox"/> Hinterland | <input checked="" type="checkbox"/> Initialization cells |
| <input checked="" type="checkbox"/> ipyparallel/main | <input type="checkbox"/> isort formatter | <input checked="" type="checkbox"/> jupyter-js-widgets/extension | <input type="checkbox"/> Keyboard shortcut editor |
| <input type="checkbox"/> Launch QTConsole | <input type="checkbox"/> Limit Output | <input type="checkbox"/> Live Markdown Preview | <input type="checkbox"/> Load TeX macros |
| <input type="checkbox"/> Move selected cells | <input type="checkbox"/> Navigation-Hotkeys | <input checked="" type="checkbox"/> Nbextensions dashboard tab | <input checked="" type="checkbox"/> Nbextensions edit menu item |
| <input type="checkbox"/> nbTranslate | <input type="checkbox"/> Notify | <input type="checkbox"/> Printview | <input type="checkbox"/> Python Markdown |
| <input type="checkbox"/> Rubberband | <input type="checkbox"/> Ruler | <input type="checkbox"/> Ruler in Editor | <input type="checkbox"/> Runtools |
| <input type="checkbox"/> Scratchpad | <input type="checkbox"/> ScrollDown | <input type="checkbox"/> Select CodeMirror Keymap | <input type="checkbox"/> SKILL Syntax |
| <input checked="" type="checkbox"/> Skip-Traceback | <input type="checkbox"/> Snippets | <input type="checkbox"/> Snippets Menu | <input type="checkbox"/> spellchecker |
| <input type="checkbox"/> Split Cells Notebook | <input checked="" type="checkbox"/> Table of Contents (2) | <input type="checkbox"/> table_beautifier | <input type="checkbox"/> Toggle all line numbers |
| <input type="checkbox"/> Tree Filter | <input type="checkbox"/> Variable Inspector | <input type="checkbox"/> zenmode | |

Collapsible Headings + Table of Contents (2)

Table of Contents

[1 k-Nearest Neighbor \(kNN\) exercise](#)

[2 Cross-validation](#)

▼ 1 k-Nearest Neighbor (kNN) exercise

Complete and hand in this completed worksheet (including its outputs and any supporting code outside of the worksheet) with your assignment submission. For more details see the [assignments page](#) on the course website.

The kNN classifier consists of two stages:

- During training, the classifier takes the training data and simply remembers it
- During testing, kNN classifies every test image by comparing to all training images and transferring the labels of the k most similar training examples
- The value of k is cross-validated

In this exercise you will implement these steps and understand the basic Image Classification pipeline, cross-validation, and gain proficiency in writing efficient, vectorized code.

Table of Contents

[1 k-Nearest Neighbor \(kNN\) exercise](#)

[2 Cross-validation](#)

▶ 1 k-Nearest Neighbor (kNN) exercise [...]

▶ 2 Cross-validation [...]

Hinterland

- Code hints without “Tab”

```
pipeline, cross_validation, etc
```

```
In [ ]: 1 from num
```

numbers

```
In [1]: 1 # Run setup c
```


numpy

Skip-Traceback + ExecuteTime


- Hide error tracebacks and warnings.

```
1 # Run some setup code for this notebook.
2 from __future__ import print_function
3
4 import random
5 import numpy as np
6 from cs231n.data_utils import load_CIFAR10
7 import matplotlib.pyplot as plt
```

executed in 12ms, finished 00:00:20 2019-10-01

 ModuleNotFoundError: No module named 'numpy' ▶

executed in 12ms, finished 00:00:20 2019-10-01

 ModuleNotFoundError: No module named 'numpy' ▼

ModuleNotFoundError Traceback
k (most recent call last)

<ipython-input-2-5ab60b3e0ac7> in <module>

3

4 import random

----> 5 import numpy as np

6 from cs231n.data_utils import load_CIFAR10

7 import matplotlib.pyplot as plt

ModuleNotFoundError: No module named 'numpy'

Add Wolfram Engine to Jupyter

- Download and install free Wolfram Engine for Developers
<https://www.wolfram.com/engine/>
- Visit project page
<https://github.com/WolframResearch/WolframLanguageForJupyter>
- Run in wolframscript:
configure-jupyter.wls add

