Be prepared to enter the world of AI

Environment setup guide

- Minchang (Carson) Zhang

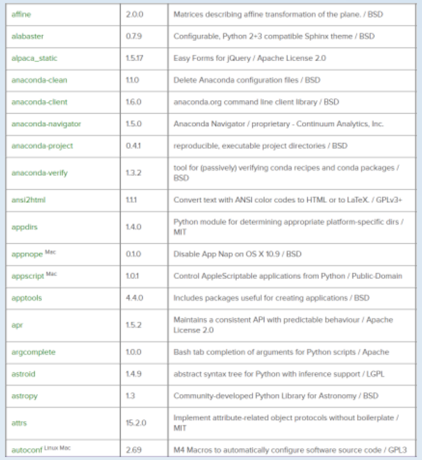
Table of Content

- Virtual environment setup on Windows OS
- Tensorflow Installation
- Quick introduction to the fundamental libraries
- An important tool: Jupyter Notebook
- Summary

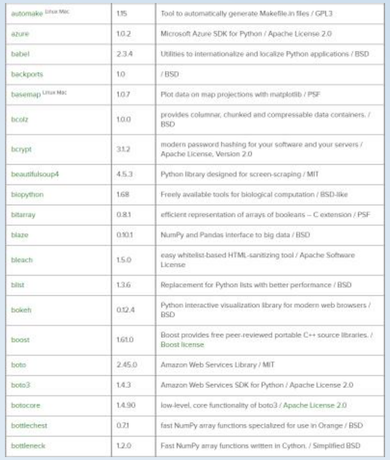
Virtual environment setup on Windows OS

We choose Anaconda over Virtualenv, why? BUILT-IN packages!!

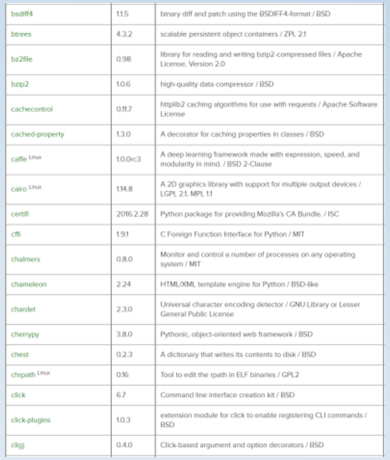
- Over 150 packages are automatically installed with Anaconda.



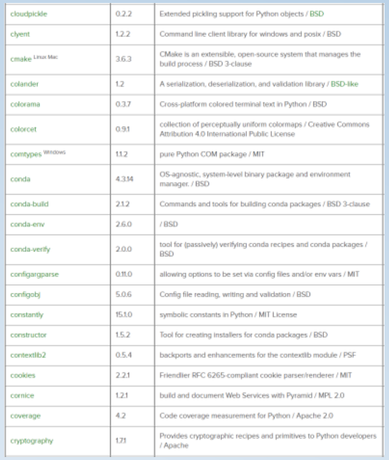
affine	2.0.0	Matrices describing affine transformation of the plane. / BSD
alabaster	0.7.9	Configurable, Python 2+3 compatible Sphinx theme / BSD
alpaca_static	1.5.17	Easy Forms for jQuery / Apache License 2.0
anaconda-clean	1.1.0	Delete Anaconda configuration files / BSD
anaconda-client	1.6.0	anaconda.org command line client library / BSD
anaconda-navigator	1.5.0	Anaconda Navigator / proprietary - Continuum Analytics, Inc.
anaconda-project	0.4.1	reproducible, executable project directories / BSD
anaconda-verify	1.3.2	tool for (passively) verifying conda recipes and conda packages / BSD
ansi2html	1.1.1	Convert text with ANSI color codes to HTML or to LaTeX. / GPLv3+
appdirs	1.4.0	Python module for determining appropriate platform-specific dirs / MIT
appnope Mac	0.1.0	Disable App Nap on OS X 10.9 / BSD
appscript Mac	1.0.1	Control AppleScriptable applications from Python / Public-Domain
apptools	4.4.0	Includes packages useful for creating applications / BSD
apr	1.5.2	Maintains a consistent API with predictable behaviour / Apache License 2.0
argcomplete	1.0.0	Bash tab completion of arguments for Python scripts / Apache
astroid	1.4.9	abstract syntax tree for Python with inference support / LGPL
astropy	1.3	Community-developed Python Library for Astronomy / BSD
attrs	15.2.0	Implement attribute-related object protocols without boilerplate / MIT
autoconf Linux Mac	2.69	M4 Macros to automatically configure software source code / GPL3



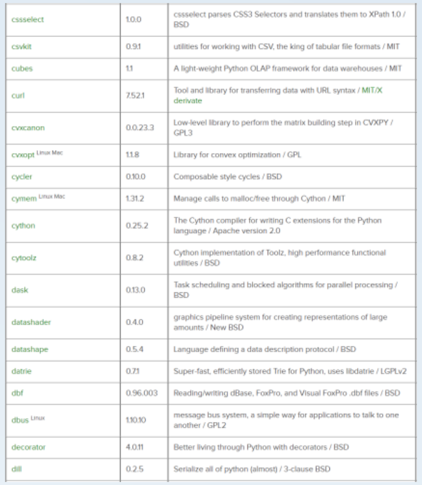
automake Linux Mac	1.15	Tool to automatically generate Makefile.in files / GPL3
azure	1.0.2	Microsoft Azure SDK for Python / Apache License 2.0
babel	2.3.4	Utilities to internationalize and localize Python applications / BSD
backports	1.0	/ BSD
basemap Linux Mac	1.0.7	Plot data on map projections with matplotlib / PSF
bcolz	1.0.0	provides columnar, chunked and compressable data containers. / BSD
bcrypt	3.1.2	modern password hashing for your software and your servers / Apache License, Version 2.0
beautifulsoup4	4.5.3	Python library designed for screen-scraping / MIT
biopython	1.68	Freely available tools for biological computation / BSD-like
bitarray	0.8.1	efficient representation of arrays of booleans – C extension / PSF
blaze	0.10.1	NumPy and Pandas interface to big data / BSD
bleach	1.5.0	easy whitelist-based HTML-sanitizing tool / Apache Software License
blist	1.3.6	Replacement for Python lists with better performance / BSD
bokeh	0.12.4	Python interactive visualization library for modern web browsers / BSD
boost	1.61.0	Boost provides free peer-reviewed portable C++ source libraries. / Boost license
boto	2.45.0	Amazon Web Services Library / MIT
boto3	1.4.3	Amazon Web Services SDK for Python / Apache License 2.0
botocore	1.4.90	low-level, core functionality of boto3 / Apache License 2.0
bottlechest	0.7.1	fast NumPy array functions specialized for use in Orange / BSD
bottleneck	1.2.0	Fast NumPy array functions written in Cython. / Simplified BSD



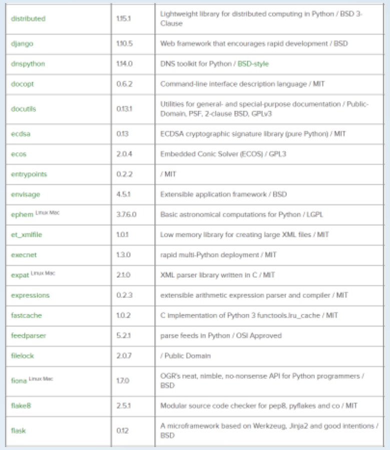
bsdiff4	1.1.5	binary diff and patch using the BSDIFF4-format / BSD
btrees	4.3.2	scalable persistent object containers / ZPL 2.1
bz2file	0.98	library for reading and writing bzip2-compressed files / Apache License, Version 2.0
bzip2	1.0.6	high-quality data compressor / BSD
cachecontrol	0.11.7	httplib2 caching algorithms for use with requests / Apache Software License
cached-property	1.3.0	A decorator for caching properties in classes / BSD
caffe Linux	1.0.0rc3	A deep learning framework made with expression, speed, and modularity in mind. / BSD 2-Clause
cairo Linux	1.14.8	A 2D graphics library with support for multiple output devices / LGPL 2.1, MPL 1.1
certifi	2016.2.28	Python package for providing Mozilla's CA Bundle. / ISC
cffi	1.9.1	C Foreign Function Interface for Python / MIT
chalmers	0.8.0	Monitor and control a number of processes on any operating system / MIT
chameleon	2.24	HTML/XML template engine for Python / BSD-like
chardet	2.3.0	Universal character encoding detector / GNU Library or Lesser General Public License
cherrypy	3.8.0	Pythonic, object-oriented web framework / BSD
chest	0.2.3	A dictionary that writes its contents to disk / BSD
chrpath Linux	0.16	Tool to edit the rpath in ELF binaries / GPL2
click	6.7	Command line interface creation kit / BSD
click-plugins	1.0.3	extension module for click to enable registering CLI commands / BSD
cligj	0.4.0	Click-based argument and option decorators / BSD



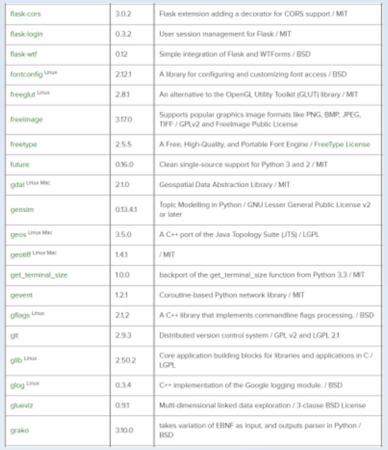
cloudpickle	0.2.2	Extended pickling support for Python objects / BSD
clyent	1.2.2	Command line client library for windows and posix / BSD
cmake Linux Mac	3.6.3	CMake is an extensible, open-source system that manages the build process / BSD 3-clause
colander	1.2	A serialization, deserialization, and validation library / BSD-like
colorama	0.3.7	Cross-platform colored terminal text in Python / BSD
colorcet	0.9.1	collection of perceptually uniform colormaps / Creative Commons Attribution 4.0 International Public License
comtypes Windows	1.1.2	pure Python COM package / MIT
conda	4.3.14	OS-agnostic, system-level binary package and environment manager. / BSD
conda-build	2.1.2	Commands and tools for building conda packages / BSD 3-clause
conda-env	2.6.0	/ BSD
conda-verify	2.0.0	tool for (passively) verifying conda recipes and conda packages / BSD
configargparse	0.11.0	allowing options to be set via config files and/or env vars / MIT
configobj	5.0.6	Config file reading, writing and validation / BSD
constantly	15.1.0	symbolic constants in Python / MIT License
constructor	1.5.2	Tool for creating installers for conda packages / BSD
contextlib2	0.5.4	backports and enhancements for the contextlib module / PSF
cookies	2.2.1	Friendlier RFC 6265-compliant cookie parser/renderer / MIT
cornice	1.2.1	build and document Web Services with Pyramid / MPL 2.0
coverage	4.2	Code coverage measurement for Python / Apache 2.0
cryptography	1.7.1	Provides cryptographic recipes and primitives to Python developers / Apache



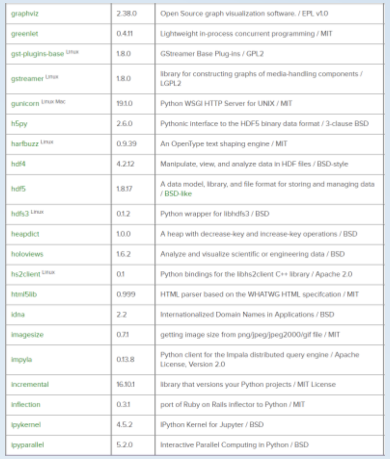
cssselect	1.0.0	cssselect parses CSS3 Selectors and translates them to XPath 1.0 / BSD
csvkit	0.9.1	utilities for working with CSV, the king of tabular file formats / MIT
cubes	1.1	A light-weight Python OLAP framework for data warehouses / MIT
curl	7.52.1	Tool and library for transferring data with URL syntax / MIT/X derivate
cvxcanon	0.0.23.3	Low-level library to perform the matrix building step in CVXPY / GPL3
cvxopt Linux Mac	1.1.8	Library for convex optimization / GPL
cycler	0.10.0	Composable style cycles / BSD
cymem Linux Mac	1.31.2	Manage calls to malloc/free through Cython / MIT
cython	0.25.2	The Cython compiler for writing C extensions for the Python language / Apache version 2.0
cytoolz	0.8.2	Cython implementation of Toolz, high performance functional utilities / BSD
dask	0.13.0	Task scheduling and blocked algorithms for parallel processing / BSD
datashader	0.4.0	graphics pipeline system for creating representations of large amounts / New BSD
datashape	0.5.4	Language defining a data description protocol / BSD
datrie	0.7.1	Super-fast, efficiently stored Trie for Python, uses libdatrie / LGPLv2
dbf	0.96.003	Reading/writing dBase, FoxPro, and Visual FoxPro .dbf files / BSD
dbus Linux	1.10.10	message bus system, a simple way for applications to talk to one another / GPL2
decorator	4.0.11	Better living through Python with decorators / BSD
dill	0.2.5	Serialize all of python (almost) / 3-clause BSD



distributed	1.15.1	Lightweight library for distributed computing in Python / BSD 3-Clause
django	1.10.5	Web framework that encourages rapid development / BSD
dnspython	1.14.0	DNS toolkit for Python / BSD-style
docopt	0.6.2	Command-line interface description language / MIT
docutils	0.13.1	Utilities for general- and special-purpose documentation / Public-Domain, PSF, 2-clause BSD, GPLv3
ecdsa	0.13	ECDSA cryptographic signature library (pure Python) / MIT
ecos	2.0.4	Embedded Conic Solver (ECOS) / GPL3
entrypoints	0.2.2	/ MIT
envisage	4.5.1	Extensible application framework / BSD
ephem Linux Mac	3.7.6.0	Basic astronomical computations for Python / LGPL
et_xmlfile	1.0.1	Low memory library for creating large XML files / MIT
execnet	1.3.0	rapid multi-Python deployment / MIT
expat Linux Mac	2.1.0	XML parser library written in C / MIT
expressions	0.2.3	extensible arithmetic expression parser and compiler / MIT
fastcache	1.0.2	C implementation of Python 3 functools.lru_cache / MIT
feedparser	5.2.1	parse feeds in Python / OSI Approved
filelock	2.0.7	/ Public Domain
fiona Linux Mac	1.7.0	OGR's neat, nimble, no-nonsense API for Python programmers / BSD
flake8	2.5.1	Modular source code checker for pep8, pyflakes and co / MIT
flask	0.12	A microframework based on Werkzeug, Jinja2 and good intentions / BSD



flask-cors	3.0.2	Flask extension adding a decorator for CORS support / MIT
flask-login	0.3.2	User session management for Flask / MIT
flask-wtf	0.12	Simple integration of Flask and WTForms / BSD
fontconfig Linux	2.12.1	A library for configuring and customizing font access / BSD
freeglut Linux	2.8.1	An alternative to the OpenGL Utility Toolkit (GLUT) library / MIT
freeimage	3.17.0	Supports popular graphics image formats like PNG, BMP, JPEG, TIFF / GPLv2 and FreeImage Public License
freetype	2.5.5	A Free, High-Quality, and Portable Font Engine / FreeType License
future	0.16.0	Clean single-source support for Python 3 and 2 / MIT
gdal Linux Mac	2.1.0	Geospatial Data Abstraction Library / MIT
gensim	0.13.4.1	Topic Modelling in Python / GNU Lesser General Public License v2 or later
geos Linux Mac	3.5.0	A C++ port of the Java Topology Suite (JTS) / LGPL
geotiff Linux Mac	1.4.1	/ MIT
get_terminal_size	1.0.0	backport of the get_terminal_size function from Python 3.3 / MIT
gevent	1.2.1	Coroutine-based Python network library / MIT
gflags Linux	2.1.2	A C++ library that implements commandline flags processing. / BSD
git	2.9.3	Distributed version control system / GPL v2 and LGPL 2.1
glib Linux	2.50.2	Core application building blocks for libraries and applications in C / LGPL
glog Linux	0.3.4	C++ implementation of the Google logging module. / BSD
glueviz	0.9.1	Multi-dimensional linked data exploration / 3-clause BSD License
grako	3.10.0	takes variation of EBNF as input, and outputs parser in Python / BSD

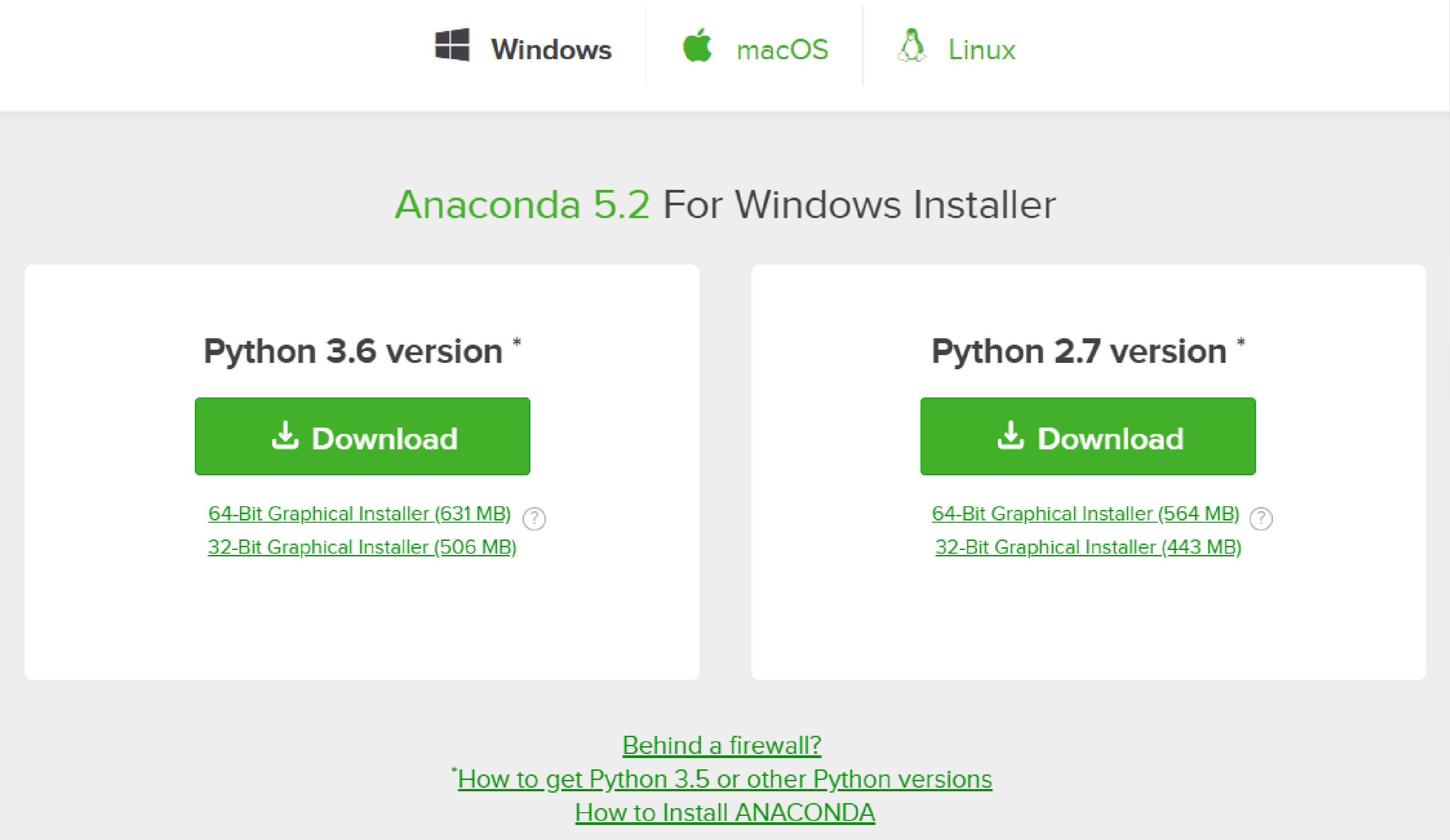


graphviz	2.38.0	Open Source graph visualization software. / EPL v1.0
greenlet	0.4.11	Lightweight in-process concurrent programming / MIT
gst-plugins-base Linux	1.8.0	GStreamer Base Plug-ins / GPL2
gstreamer Linux	1.8.0	library for constructing graphs of media-handling components / LGPL2
gunicorn Linux Mac	19.1.0	Python WSGI HTTP Server for UNIX / MIT
h5py	2.6.0	Pythonic interface to the HDF5 binary data format / 3-clause BSD
harfbuzz Linux	0.9.39	An OpenType text shaping engine / MIT
hdf4	4.2.12	Manipulate, view, and analyze data in HDF files / BSD-style
hdf5	1.8.17	A data model, library, and file format for storing and managing data / BSD-like
hdfs3 Linux	0.1.2	Python wrapper for libhdfs3 / BSD
heapdict	1.0.0	A heap with decrease-key and increase-key operations / BSD
holoviews	1.6.2	Analyze and visualize scientific or engineering data / BSD
hs2client Linux	0.1	Python bindings for the libhs2client C++ library / Apache 2.0
html5lib	0.999	HTML parser based on the WHATWG HTML specifcation / MIT
idna	2.2	Internationalized Domain Names in Applications / BSD
imagesize	0.7.1	getting image size from png/jpeg/jpeg2000/gif file / MIT
impyla	0.13.8	Python client for the Impala distributed query engine / Apache License, Version 2.0
incremental	16.10.1	library that versions your Python projects / MIT License
inflection	0.3.1	port of Ruby on Rails inflector to Python / MIT
ipykernel	4.5.2	IPython Kernel for Jupyter / BSD
ipyparallel	5.2.0	Interactive Parallel Computing in Python / BSD

Virtual environment setup on Windows OS

Install Anaconda:

- Choose the version that you use most frequently



Anaconda 5.2 For Windows Installer

Python 3.6 version *

64-Bit Graphical Installer (631 MB)

32-Bit Graphical Installer (506 MB)

Python 2.7 version *

64-Bit Graphical Installer (564 MB)

32-Bit Graphical Installer (443 MB)

Behind a firewall?

*How to get Python 3.5 or other Python versions

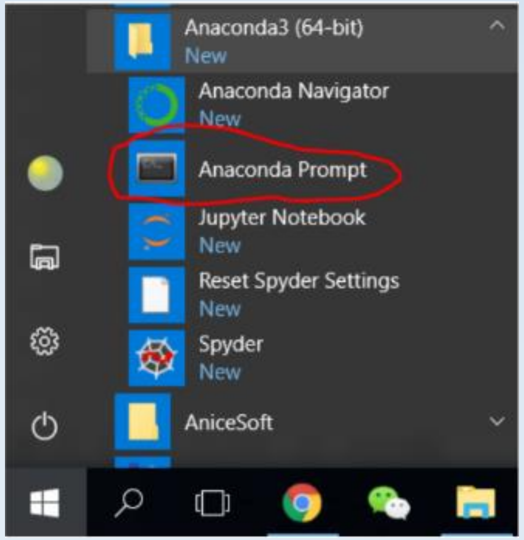
How to Install ANACONDA

Virtual environment setup on Windows OS

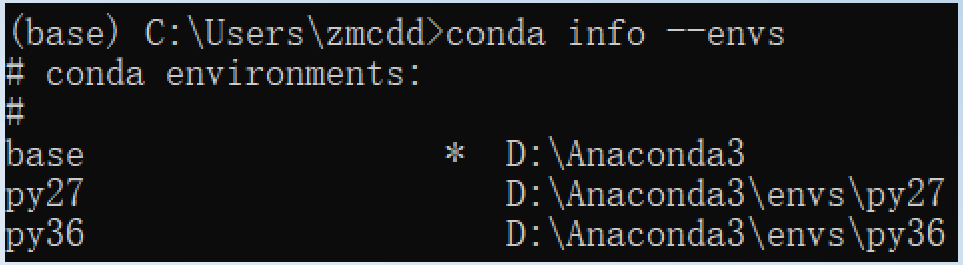
Virtual environment setup on Windows OS

Setup python environment with Anaconda

- Use Anaconda Prompt (NOT the windows Prompt)

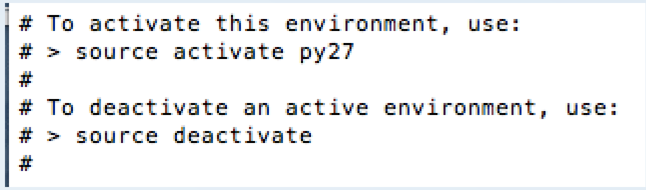


- conda create –n environment_name
  - conda create –n new_environment_name --clone existing_env_name
  - conda create –n environment_name python=2.7
  - Check existing environment: conda info --envs



```
(base) C:\Users\zmcdd>conda info --envs
# conda environments:
#
base                  *  D:\Anaconda3
py27                     D:\Anaconda3\envs\py27
py36                     D:\Anaconda3\envs\py36
```

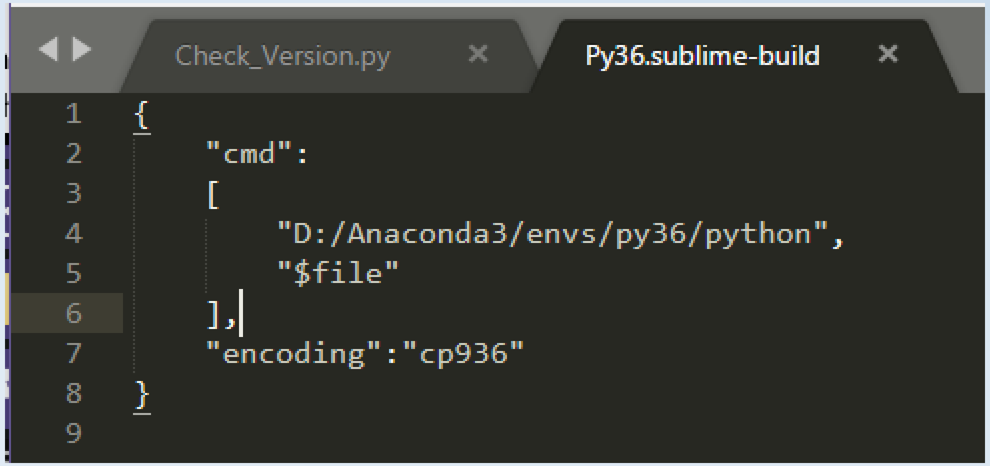
- Working within the environment
  - on Windows: activate environment_name (deactivate)
  - On Mac: source activate environment_name (source deactivate)



```
# To activate this environment, use:
# > source activate py27
#
# To deactivate an active environment, use:
# > source deactivate
#
```

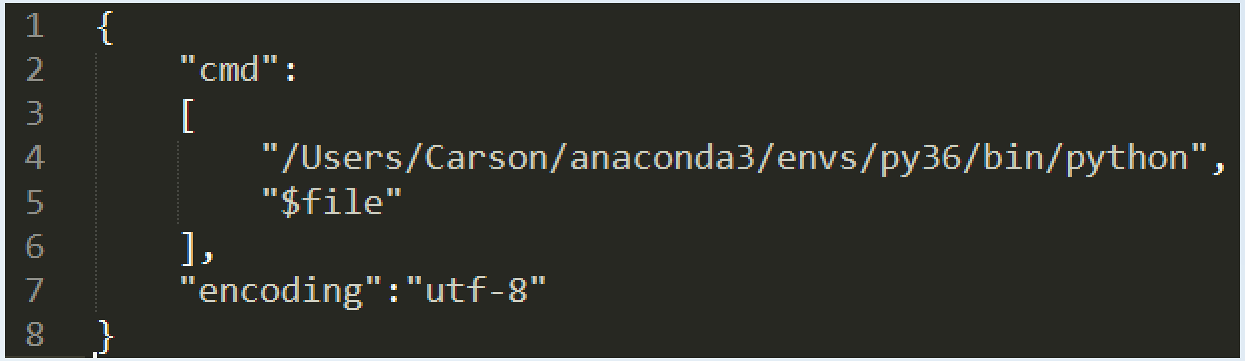
Sublime Text:

- Create new built system



```
{
    "cmd":
    [
        "D:/Anaconda3/envs/py36/python",
        "$file"
    ],
    "encoding":"cp936"
}
```

Windows



```
{
    "cmd":
    [
        "/Users/Carson/anaconda3/envs/py36/bin/python",
        "$file"
    ],
    "encoding":"utf-8"
}
```

Mac

Virtual environment setup on Windows OS

Some useful conda commands:

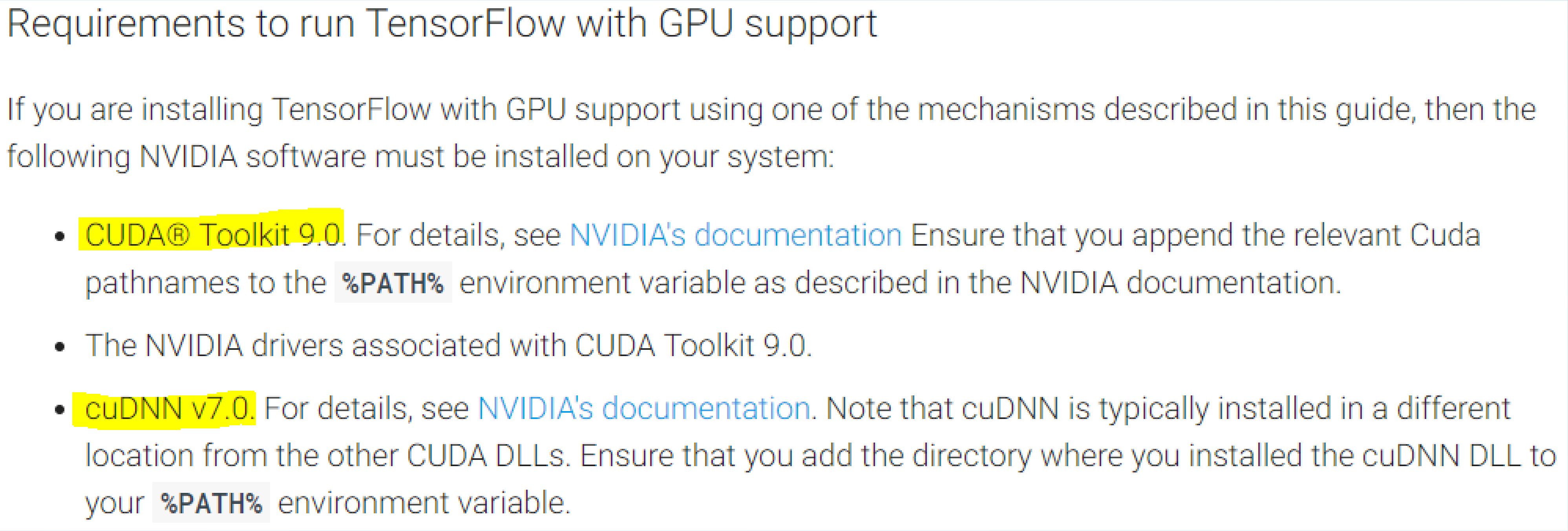
Commands	Description
conda list	list all the packages installed
conda info –envs	list all the environment created
conda create -n *environment_name	create an environment
conda create –name *name1 –clone *name2	create a new environment (name1) to be a clone of another (name2) one
conda install *package1 *package2	install multipe packages at the same time
conda install *package = version	install a package with a particular version

Virtual environment setup on Windows OS

Tensorflow Installation

Tensorflow Installation

1. Activate virtual environment: activate environment_name
   - For example: activate Tensorflow
2. Install Tensorflow (GPU Version): pip install tensorflow-gpu
   - No GPU version: pip install tensorflow
   - IMPORTANT: check the installed version of tensorflow



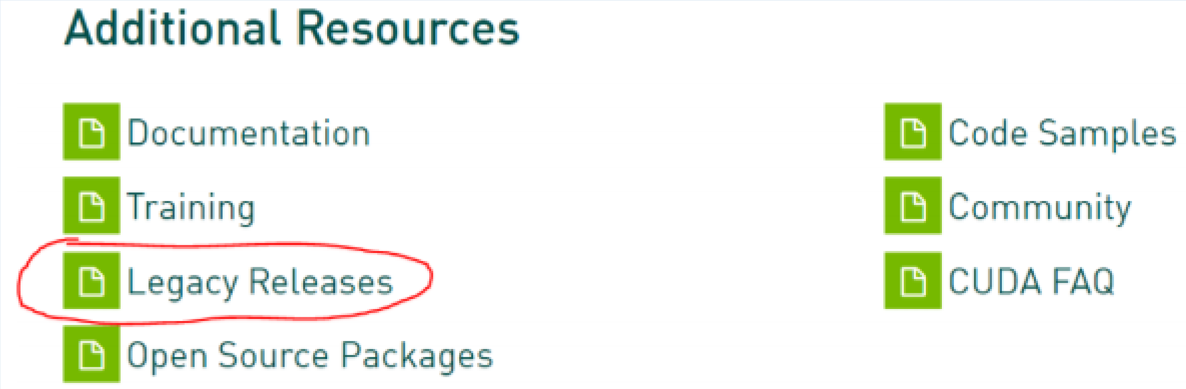
Requirements to run TensorFlow with GPU support

If you are installing TensorFlow with GPU support using one of the mechanisms described in this guide, then the following NVIDIA software must be installed on your system:

- CUDA® Toolkit 9.0. For details, see NVIDIA's documentation Ensure that you append the relevant Cuda pathnames to the `%PATH%` environment variable as described in the NVIDIA documentation.
- The NVIDIA drivers associated with CUDA Toolkit 9.0.
- cuDNN v7.0. For details, see NVIDIA's documentation. Note that cuDNN is typically installed in a different location from the other CUDA DLLs. Ensure that you add the directory where you installed the cuDNN DLL to your `%PATH%` environment variable.

Tensorflow Installation

1. Activate virtual environment: activate environment_name
    - For example: activate Tensorflow
2. Install Tensorflow (GPU Version): pip install tensorflow-gpu
    - No GPU version: pip install tensorflow
    - IMPORTANT: check the installed version of tensorflow
3. Install CUDA
    - Trick 1: Check the legacy release

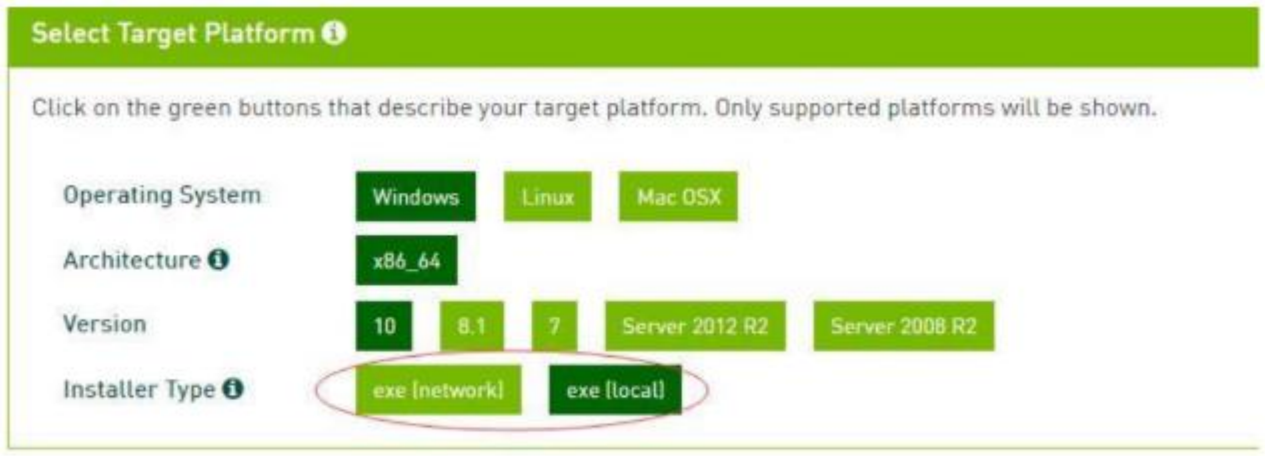


Additional Resources

- Documentation
- Training
- Legacy Releases
- Open Source Packages
- Code Samples
- Community
- CUDA FAQ

Tensorflow Installation

1. Activate virtual environment: activate environment_name
   - For example: activate Tensorflow
2. Install Tensorflow (GPU Version): pip install tensorflow-gpu
   - No GPU version: pip install tensorflow
   - IMPORTANT: check the installed version of tensorflow
3. Install CUDA
   - Trick 1: Check the legacy release
   - Trick 2: Download local version



Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown.

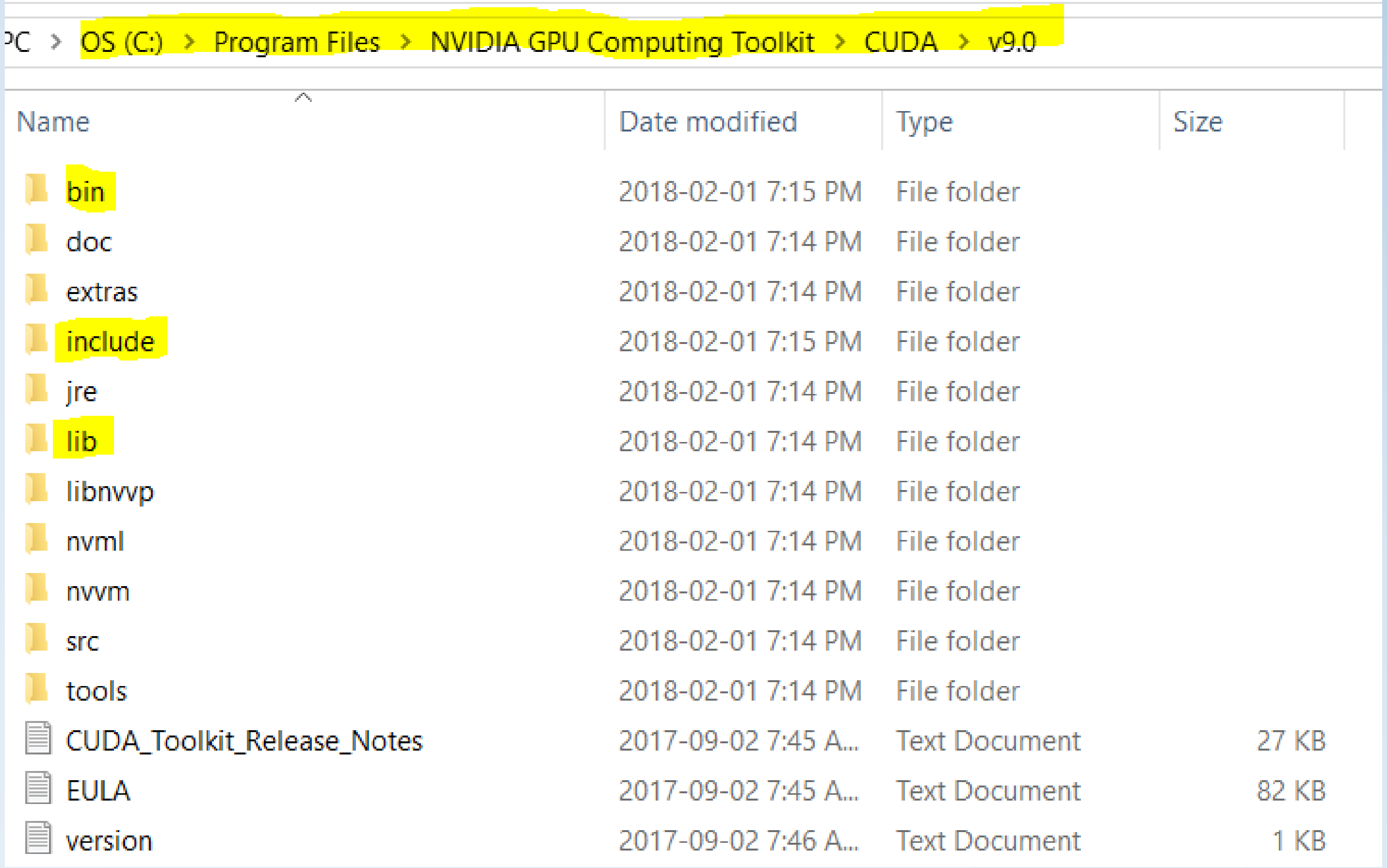
Operating System: [✓ Windows] Linux Mac OSX

Architecture: [✓ x86_64]

Version: [✓ 10] 8.1 7 Server 2012 R2 Server 2008 R2

Installer Type: exe (network) [✓ exe (local)]

1. Activate virtual environment: activate environment_name
    - For example: activate Tensorflow
2. Install Tensorflow (GPU Version): pip install tensorflow-gpu
    - No GPU version: pip install tensorflow
    - IMPORTANT: check the installed version of tensorflow
3. Install CUDA
    - Trick 1: Check the legacy release
    - Trick 2: Download local version
    - Trick 3: Use Edge!!
4. Download and use cuDnn
    - Copy from download and paste to installed CUDA directory



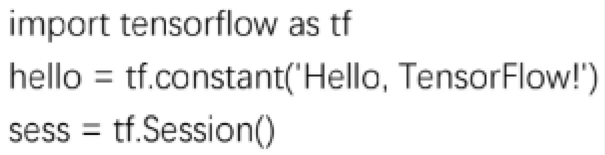
PC > OS (C:) > Program Files > NVIDIA GPU Computing Toolkit > CUDA > v9.0

Name	Date modified	Type	Size
bin	2018-02-01 7:15 PM	File folder	
doc	2018-02-01 7:14 PM	File folder	
extras	2018-02-01 7:14 PM	File folder	
include	2018-02-01 7:15 PM	File folder	
jre	2018-02-01 7:14 PM	File folder	
lib	2018-02-01 7:14 PM	File folder	
libnvvp	2018-02-01 7:14 PM	File folder	
nvml	2018-02-01 7:14 PM	File folder	
nvvm	2018-02-01 7:14 PM	File folder	
src	2018-02-01 7:14 PM	File folder	
tools	2018-02-01 7:14 PM	File folder	
CUDA_Toolkit_Release_Notes	2017-09-02 7:45 A...	Text Document	27 KB
EULA	2017-09-02 7:45 A...	Text Document	82 KB
version	2017-09-02 7:46 A...	Text Document	1 KB

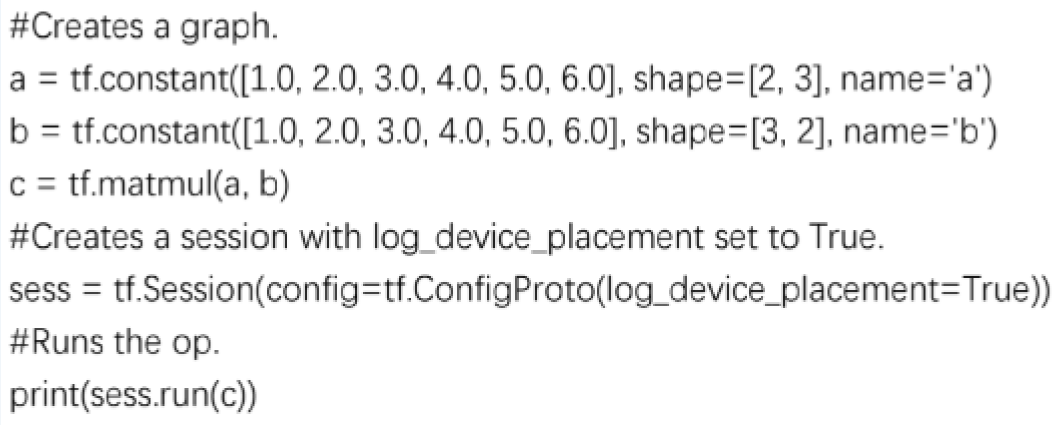
Tensorflow Installation

Tensorflow Installation

1. Activate virtual environment: activate environment_name
   - For example: activate Tensorflow
2. Install Tensorflow (GPU Version): pip install tensorflow-gpu
   - No GPU version: pip install tensorflow
   - IMPORTANT: check the installed version of tensorflow
3. Install CUDA
   - Trick 1: Check the legacy release
   - Trick 2: Download local version
   - Trick 3: Use Edge!!
4. Download and use cuDnn
   - Copy from download and paste to installed CUDA directory
5. Test code



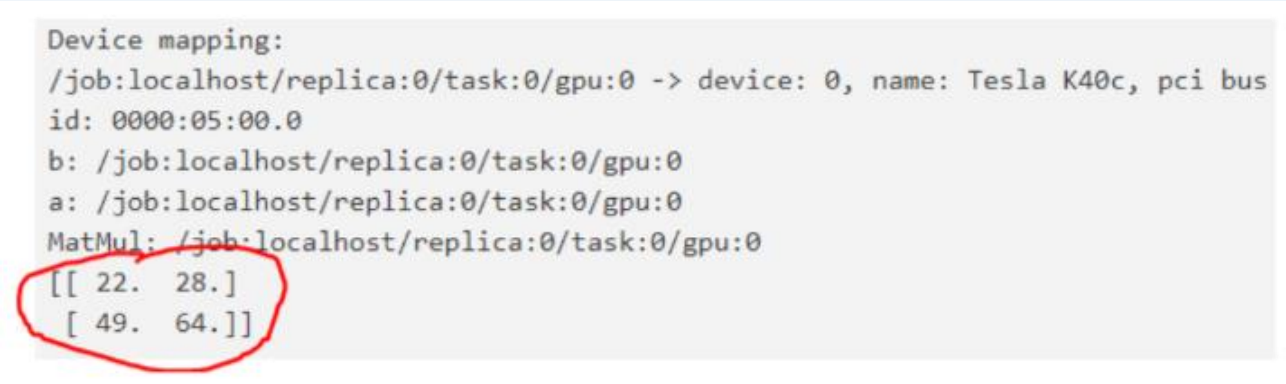
```
import tensorflow as tf
hello = tf.constant('Hello, TensorFlow!')
sess = tf.Session()
```



```
#Creates a graph.
a = tf.constant([1.0, 2.0, 3.0, 4.0, 5.0, 6.0], shape=[2, 3], name='a')
b = tf.constant([1.0, 2.0, 3.0, 4.0, 5.0, 6.0], shape=[3, 2], name='b')
c = tf.matmul(a, b)
#Creates a session with log_device_placement set to True.
sess = tf.Session(config=tf.ConfigProto(log_device_placement=True))
#Runs the op.
print(sess.run(c))
```

Tensorflow Installation

1. Activate virtual environment: activate environment_name
    - For example: activate Tensorflow
2. Install Tensorflow (GPU Version): pip install tensorflow-gpu
    - No GPU version: pip install tensorflow
    - IMPORTANT: check the installed version of tensorflow
3. Install CUDA
    - Trick 1: Check the legacy release
    - Trick 2: Download local version
    - Trick 3: Use Edge!!
4. Download and use cuDnn
    - Copy from download and paste to installed CUDA directory
5. Test code



```
Device mapping:
/job:localhost/replica:0/task:0/gpu:0 -> device: 0, name: Tesla K40c, pci bus
id: 0000:05:00.0
b: /job:localhost/replica:0/task:0/gpu:0
a: /job:localhost/replica:0/task:0/gpu:0
MatMul: /job:localhost/replica:0/task:0/gpu:0
[[ 22.  28.]
 [ 49.  64.]]
```

Quick introduction to the fundamental libraries

Quick introduction to the fundamental libraries

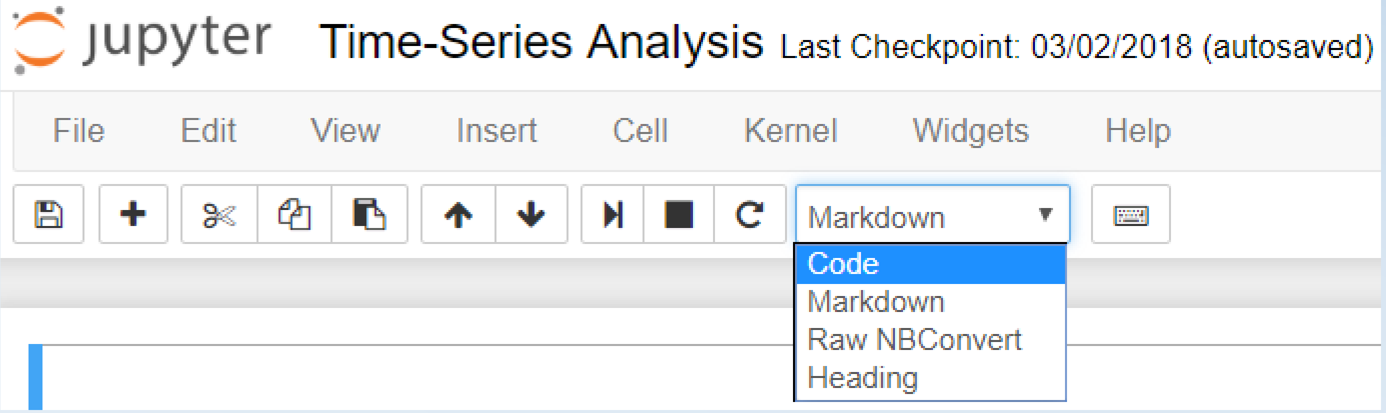
1. Core Libraries
   1. Numpy: advanced calculation package
   2. Scipy: built upon Numpy, focus on linear algebra, optimization, integration, and statistics
   3. Pandas: data analysis
2. Data Visualization
   1. Matplotlib: can produce many types of plots/charts/histograms
   2. Seaborn: based on Matplotlib, focuse on the visualization of statistical models
   3. Bokeh: aim at interactive visualizations
3. Machine Learning, Deep Learning, etc:
   1. Scikit-learn: essential for machine learning algorithms
   2. TensorFlow: enables lower (model-wise) level of modeling
   3. Keras: built upon TensorFlow or Theano (i.e. use them as low-level calculation source)
   4. Theano: no longer being actively maintained or developed
4. Others:
   1. NLP: NLTK, Gensim
   2. Data scraping: Scrapy, Beautifulsoup
   3. Image processing: OpenCV, Pillow

An important tool: Jupyter Notebook

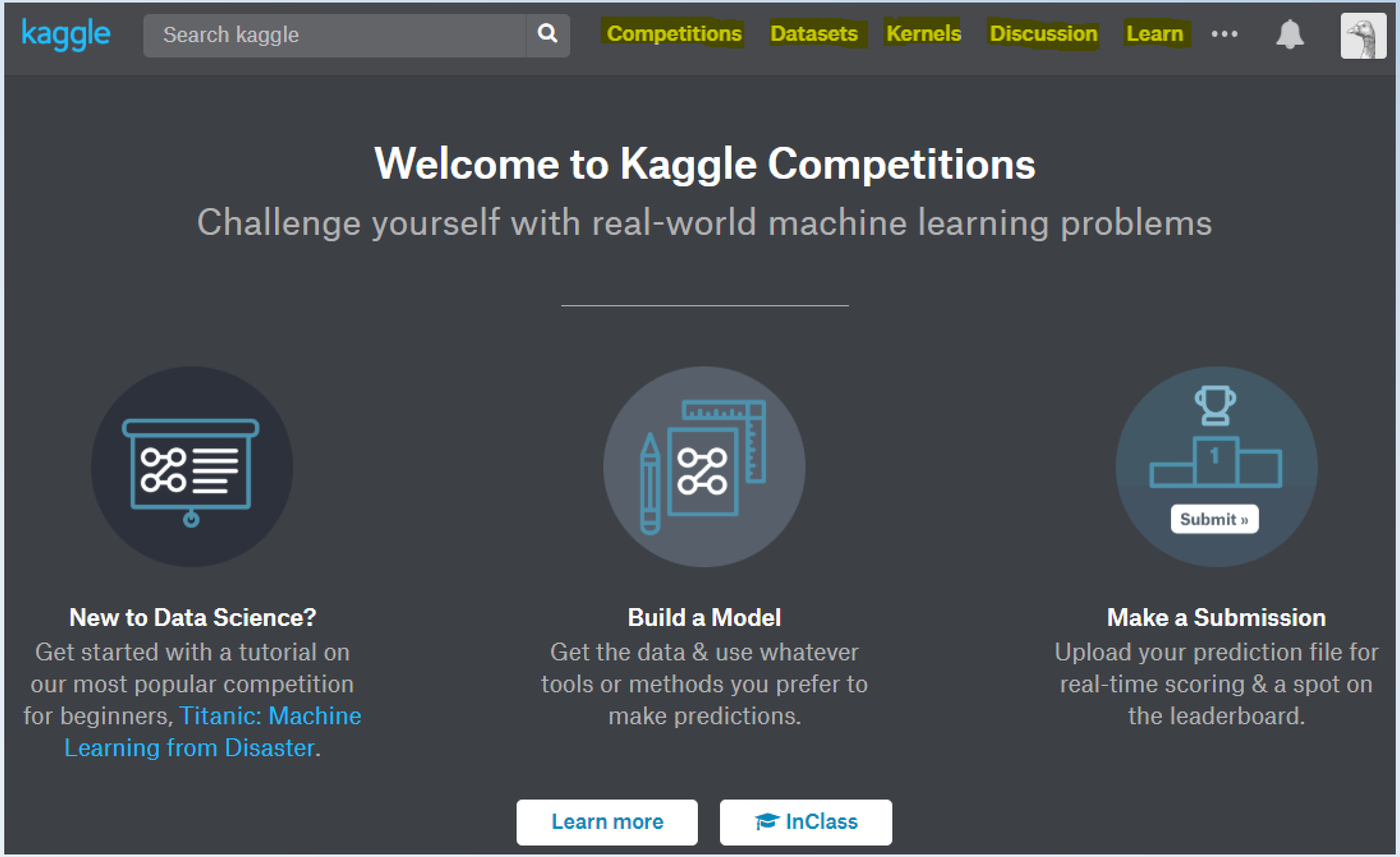
An important tool: Jupyter Notebook

- Jupyter Notebook is integrated with Anaconda

- Jupyter Notebook is very similar to ipython or Methematica


jupyter Time-Series Analysis Last Checkpoint: 03/02/2018 (autosaved)

- Kaggle!!



Welcome to Kaggle Competitions

Challenge yourself with real-world machine learning problems

New to Data Science?
Get started with a tutorial on our most popular competition for beginners, Titanic: Machine Learning from Disaster.

Build a Model
Get the data & use whatever tools or methods you prefer to make predictions.

Make a Submission
Upload your prediction file for real-time scoring & a spot on the leaderboard.

An important tool: Jupyter Notebook

Summary

- Virtual environment
  - Anaconda
- Tensorflow Installation
  - GPU version
  - CUDA
  - cuDnn
- Fundamental libraries
  - Numpy
  - Pandas
  - Scikit-learn
- Jupyter Notebook and Kaggle

QUESTIONS ?