

# Tutorial 1

## Intro to Python + Tensorflow

CS587, Spring Semester 2017, CSD, UOC

# Outline

- Introductory notes for Python, Anaconda, Jupyter Notebook
- Notes on installing the Python/Tensorflow environment & tools
  - Anaconda (Python, Jupyter, NumPy, SciPy, Matplotlib,...)
  - Tensorflow (to be added)
- Deep Learning resources



# Python

- **Python** [www.python.org](http://www.python.org)
  - is a high-level, interpreted programming language for general-purpose programming.
  - features a dynamic type system and automatic memory management,
  - supports multiple programming paradigms, (object-oriented, imperative, functional programming, and procedural styles).
  - has a large and comprehensive standard library.
- **Official tutorial:** <https://docs.python.org/3.5/tutorial/>
- **Free Online Course @ EDX - Python 3 programming basics** [LINK](#)
- **Free Online Course @ DataCamp –** [LINK](#)
- **Python Numpy Tutorial** <http://cs231n.github.io/python-numpy-tutorial/> + Jupyter Notebook file (**Recommended**)
- **An introduction to Python** [LINK](#)
- **Python for Vision Research Tutorials** [LINK](#)



# Python

## Transitioning from MATLAB to Python !!!

### 1) [LINK](#) + jupyter notebook

- This tutorial is meant as an introduction to Python's essential scientific packages: Numpy, PIL, Matplotlib, and SciPy.
- You can download this notebook by pressing the right upper corner button and test it in your machine locally.

2) <https://docs.scipy.org/doc/numpy-dev/user/numpy-for-matlab-users.html>

3) [http://scipy.github.io/old-wiki/pages/NumPy\\_for\\_Matlab\\_Users](http://scipy.github.io/old-wiki/pages/NumPy_for_Matlab_Users)

Python+Numpy tutorial: [http://www.csd.uoc.gr/~hy119/tziritas/quick\\_intro.py](http://www.csd.uoc.gr/~hy119/tziritas/quick_intro.py)

# Anaconda

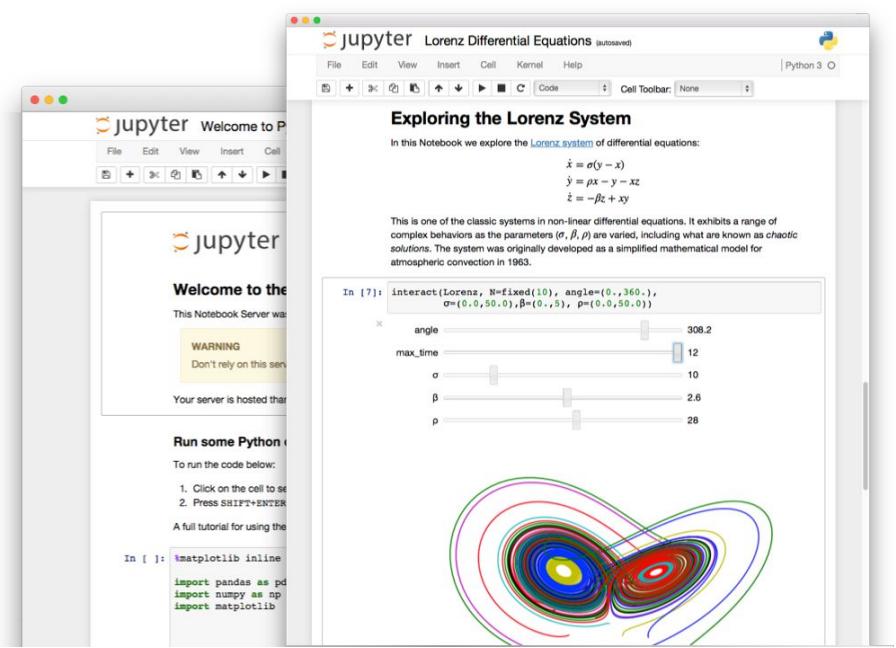


- [Anaconda](#) is a Python distribution that includes many of the most popular Python packages for science, math, engineering and data analysis.
- Includes an easy installation of Python (i.e versions 2.7, 3.4, 3.5) and over 100 pre-built and tested scientific and analytic Python packages (PIL, matplotlib, numpy, scipy, ipython,jupyter, etc).

# Jupyter (IPython) Notebook

<http://jupyter.org/>

- The Jupyter Notebook is a web application that allows you to create and share documents that contain live code, equations, visualizations and explanatory text.
- The Notebook has support for over 40 programming languages, including those popular in Data Science such as Python, R, Julia and Scala.
- Jupyter Notebook Tutorial  
<http://jupyter.readthedocs.io/en/latest/content-quickstart.html>  
<http://cs231n.github.io/ipython-tutorial/>



# Python IDEs

Code using your favorite Python IDE !

- Many of the IDEs allow working/manipulation of Jupyter Notebooks in their editors.

[PyCharm](#) (you can get a free student license!)

[Spyder](#)

[PyDev](#) - Python IDE for Eclipse

[Komodo](#)

[Thonny](#)

Or choose another from the [list](#) !

# CS 587 – Python/Anaconda Setup

We will use Python 3.5 based on Anaconda (v3.5 up to v4.3) distribution and IPython+/Jupyter Notebook to run our Deep Learning stuff based on the TensorFlow 1.0 framework.

1) Download & Install Anaconda for your OS. The bundle includes Python, Jupyter, Numpy, Scipy, and many many other Python packages/libraries.

<https://www.continuum.io/downloads>

- How to install  
<https://docs.continuum.io/anaconda/install>
- How to verify installation
  - You can confirm that Anaconda is installed and working by opening a terminal window (on Windows: Start - Command Prompt) and entering a command such as 'conda list'. If Anaconda is installed and working, this will display a list of installed packages and their versions.
  - Another way to confirm that Anaconda is installed and working by opening a terminal window (on Windows: Start - Command Prompt) and entering the command 'python' to run the Python shell. If Anaconda is installed and working, the version information it displays when it starts up will include "Continuum Analytics, Inc.". Use the command 'quit()' to exit the Python shell.
- Conda Cheat sheet  
[https://conda.io/docs/\\_downloads/conda-cheatsheet.pdf](https://conda.io/docs/_downloads/conda-cheatsheet.pdf)
- Python Tutorial: Anaconda - Installation and Using Conda  
<https://www.youtube.com/watch?v=YJC6ldI3hWk>

# CS 587 – Tensorflow Setup



2) <https://www.tensorflow.org/install/>

Choose the option for installing Tensorflow for your OS.

A screenshot of the TensorFlow website's 'Install' page. The page has an orange header with navigation links: TensorFlow (selected), Install (highlighted in white), Develop, API r1.0, Deploy, Extend, Resources, and Versions. To the right of the header is a search bar with a magnifying glass icon and the word 'Search', and a GitHub link. Below the header is a large orange banner with the word 'Install' in white. On the left side, there is a sidebar with a light gray background containing a list of installation links: 'Installing TensorFlow', 'Installing TensorFlow on Ubuntu', 'Installing TensorFlow on Mac OS X', 'Installing TensorFlow on Windows', 'Installing TensorFlow from Sources', and 'Transitioning to TensorFlow 1.0'. The main content area has a white background and features the title 'Installing TensorFlow' in a large, dark font. Below the title is a paragraph stating 'We have installation instructions for the following platform:' followed by a bulleted list: 'Linux', 'Mac OS X', 'Windows', and 'From source'. At the bottom of the main content area, there is a link to 'migrating from previous versions of TensorFlow to v1.0'.

## Installing TensorFlow

We have installation instructions for the following platform:

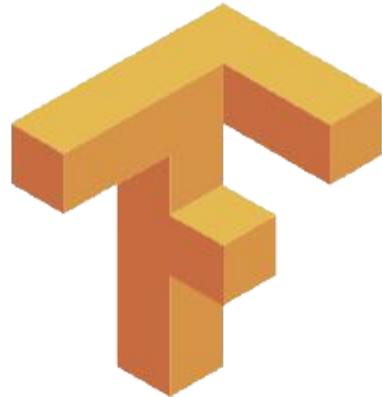
- [Linux](#)
- [Mac OS X](#)
- [Windows](#)
- [From source](#)

We also have help for [migrating from previous versions of TensorFlow to v1.0](#).

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.

Last updated February 15, 2017.

# CS 587 – Tensorflow Setup



3) Try the Tensorflow for CPU support for starters !

Recommendations for installing with Anaconda for Windows users  
and for installing with virtualenv for Ubuntu, MacOSX.

Validate your installation based on the instructions provided in the chosen page !

The screenshot shows the TensorFlow website's 'Install' section. The top navigation bar includes links for 'TensorFlow™', 'Install' (which is underlined), 'Develop', 'API r1.0', 'Deploy', 'Extend', 'Resources', 'Versions', a search bar, and a 'GitHub' link. The main content area has a yellow header with the word 'Install'. Below it, there's a sidebar with links to 'Installing TensorFlow', 'Installing TensorFlow on Ubuntu', 'Installing TensorFlow on Mac OS X', 'Installing TensorFlow on Windows' (which is highlighted in blue), 'Installing TensorFlow from Sources', and 'Transitioning to TensorFlow 1.0'. The main content area features a title 'Installing TensorFlow on Windows' and a sub-section 'Determine which TensorFlow to install'. It explains that users must choose between 'TensorFlow with CPU support only' or 'TensorFlow with GPU support'. The 'Contents' sidebar on the right lists various installation and validation topics.

**TensorFlow™** [Install](#) [Develop](#) [API r1.0](#) [Deploy](#) [Extend](#) [Resources](#) [Versions](#)  [Search](#) [GitHub](#)

**Install**

[Installing TensorFlow](#)  
[Installing TensorFlow on Ubuntu](#)  
[Installing TensorFlow on Mac OS X](#)  
**[Installing TensorFlow on Windows](#)**  
[Installing TensorFlow from Sources](#)  
[Transitioning to TensorFlow 1.0](#)

## Installing TensorFlow on Windows

This guide explains how to install TensorFlow on Windows.

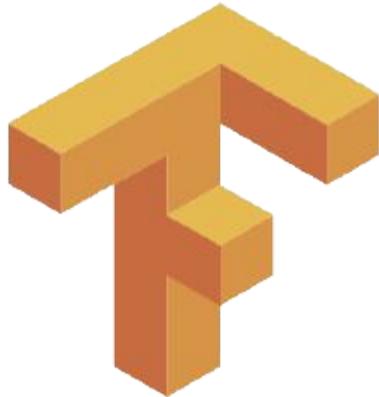
### Determine which TensorFlow to install

You must choose one of the following types of TensorFlow to install:

- **TensorFlow with CPU support only.** If your system does not have a NVIDIA® GPU, you must install this version. Note that this version of TensorFlow is typically much easier to install (typically, in 5 or 10 minutes), so even if you have an NVIDIA GPU, we recommend installing this version first.
- **TensorFlow with GPU support.** TensorFlow programs typically run significantly faster on a GPU than on a CPU. Therefore, if your system has a NVIDIA® GPU meeting the prerequisites shown below and you need to run performance-critical applications, you should ultimately install this version.

**Contents**

- Determine which TensorFlow to install
- Requirements to run TensorFlow with GPU support
- Determine how to install TensorFlow
- Installing with native pip
- Installing with Anaconda
- Validate your installation
- Common installation problems
- Stay Connected
- Support



# CS 587 – Tensorflow Setup

- Troubleshooting
  - For windows users in step 4 of “Installing with Anaconda”:  
If the “pip install –ignore....” command produces an error like  
*“...tensorflow-1.0.0-cp35-cp35m-win\_x86\_64.whl is not a supported wheel on this platform”*  
Try again by replacing the tensorflow-1.0.0-cp35-cp35m-win\_x86\_64.whl filename in the “pip install –ignore....” command with **tensorflow-1.0.0-cp35-cp35m-win\_amd64.whl**
  - More to be added...

# Tensorflow framework



- TensorFlow™ is an open source software library for numerical computation using data flow graphs.
- TensorFlow™ provides multiple APIs (app, compat, contrib, image, metrics, train, test.....).
- They are available in several languages (Python, C++, Java, Go) both for constructing and executing a TensorFlow graph.
  - Python API → the most complete and the easiest to use `import tensorflow as tf`
  - Lowest level API → TensorFlow Core `tf.placeholder`   `tf.Session`   `tf.nn`
  - High level API → `tf.contrib.learn`

# Tensors



The central unit of data in TensorFlow is the **tensor** (a set of primitive values shaped into an array of any number of dimensions).

A tensor's **rank** is its number of dimensions.

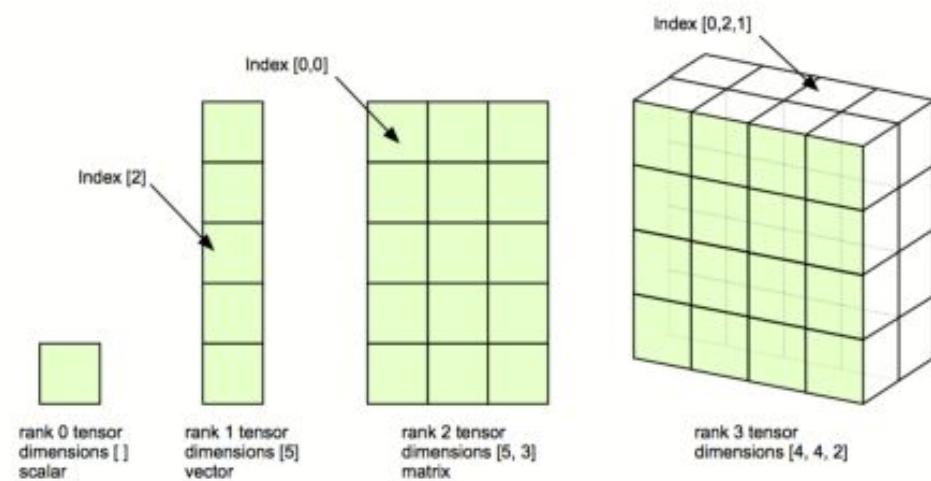
```
3 # a rank 0 tensor; this is a scalar with shape []
```

```
[1., 2., 3.] # a rank 1 tensor; this is a vector with shape [3]
```

```
[[1., 2., 3.], [4., 5., 6.]] # a rank 2 tensor; a matrix with shape [2, 3]
```

```
[[[1., 2., 3.]], [[7., 8., 9.]]] # a rank 3 tensor with shape [2, 1, 3]
```

1. Scalar:  $\mathbb{R}$
2. Vector:  $\mathbb{R}^n$
3. Matrix:  $\mathbb{R}^n \times \mathbb{R}^m$
4. 3-Tensor:  $\mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p$
5. ...

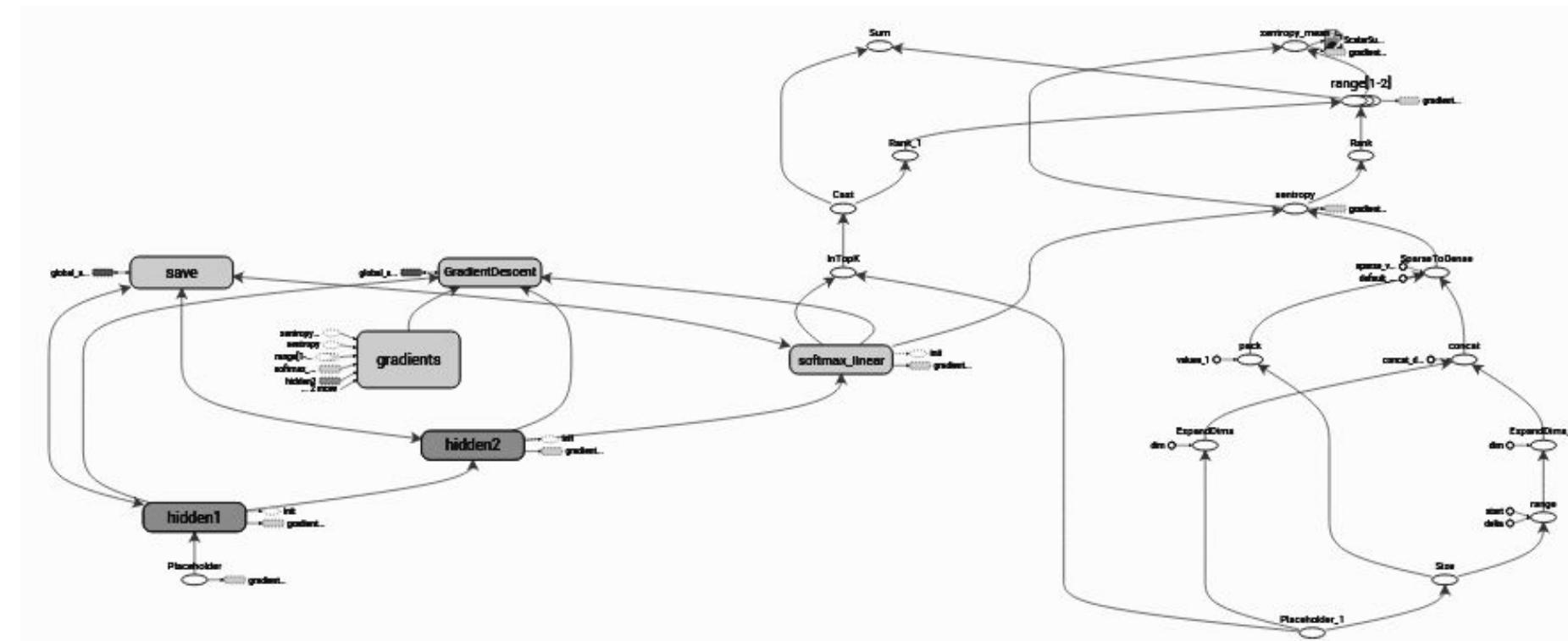


# The Computational Graph



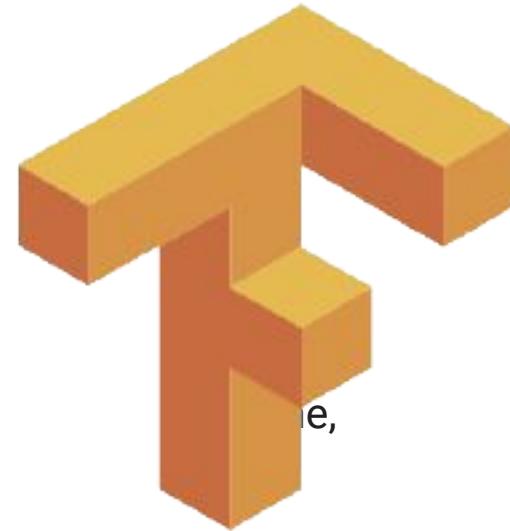
A **computational graph** is a series of TensorFlow operations arranged into a graph of nodes.

Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them.



## The Computational Graph

TensorFlow provides a symbolic graph computation API, which allows automatic differentiation of each node.



You might think of TensorFlow Core programs as consisting of two discrete sections:

- Build a computational graph
- Start a new session to evaluate the graph
  - Initialize variables
  - Execute the operations in the compiled graph

# Tensorflow support



- CPU and GPU support

```
1 with tf.Session() as sess:  
2     with tf.device("/gpu:1"):
```

where

- “/cpu:0”: The CPU of your machine.
- “/gpu:0”: The GPU of your machine, if you have one.
- “/gpu:1”: The second GPU of your machine, etc.
- etc.

```
1 from tensorflow.python.client import device_lib  
2  
3 device_lib.list_local_devices()
```

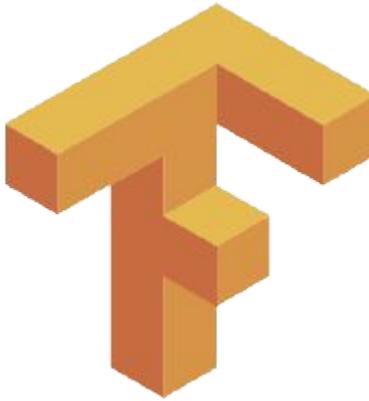
[https://www.tensorflow.org/how\\_tos/using\\_gpu/](https://www.tensorflow.org/how_tos/using_gpu/)

## Basic Concepts



- **Constants**: constant tensor
- **Placeholder variable**: allow us to feed the computational graph with numerical values in an active session at runtime.
- **Variables**: allow us to store and update parameters of our models during training
- ..
- ..
- ..

# Tensorflow tutorials



- [Guide start programming in TensorFlow](#)
- [MNIST for beginners](#)
- [Simple Tutorials using Tensorflow](#)
- [Udacity tutorials](#)
- [Awesome Tensorflow](#) (list of resources)
- [ML and DL using Python, Scikit-Learn and TensorFlow](#)
- <https://github.com/rasbt/deep-learning-book>
- Online Playground with NNs using Tensorflow → [LINK](#)

# Free Online Courses for Deep Learning

- Coursera  
Neural Networks for Machine Learning University of Toronto  
<https://www.coursera.org/learn/neural-networks/home/info>
- Udacity  
<https://www.udacity.com/course/deep-learning--ud730>
- Stanford CS231n - CS231n: Convolutional Neural Networks for Visual Recognition  
<http://cs231n.stanford.edu/index.html>

# Free Books for Deep Learning

- Neural Networks & Deep Learning Online book ([Michael Nielsen](#))  
<http://neuralnetworksanddeeplearning.com/>

Step by step introduction to NNs, DNNs + practical exercises in Python

- A beginner's guide [LINK](#)

- Deep Learning Book, MIT Press 2016 (Goodfellow, Bengio, Courville)

<http://www.deeplearningbook.org/>

Math and Machine Learning Basics + Deep Nets, Deep learning

- ANNs & DL Book <https://github.com/rasbt/deep-learning-book>
- Deep Learning Tutorial 1 <http://deeplearning.net/tutorial/deeplearning.pdf>

# Blogs + Online demos

- Deep Learning in a Nutshell (Recommended !)  
<https://devblogs.nvidia.com/parallelforall/deep-learning-nutshell-core-concepts/>
- Play with NNs online based on Tensorflow <http://playground.tensorflow.org>
- Deep Learning in your browser <http://cs.stanford.edu/people/karpathy/convnetjs/>
- colah's blog <http://colah.github.io/> (Recommended !)
- [A Beginner's Guide To Understanding Convolutional Neural Networks](#) (Recommended !)
- #FUN app : What a Deep Neural Network thinks about your #selfie  
<http://karpathy.github.io/2015/10/25/selfie/>

# Deep Learning resources

**Awesome LIST** – visit <https://github.com/ChristosChristofidis/awesome-deep-learning>  
**(or follow the links)**

- [Free Online Books](#)
- [Courses](#)
- [Videos and Lectures](#)
- [Papers](#)
- [Tutorials](#)
- [Researchers](#)
- [WebSites](#)
- [Datasets](#)
- [Frameworks](#)
- [Miscellaneous](#)

# Practical guides on CNNs

- V. Dumoulin and F. Visin, [A guide to convolution arithmetic for deep learning](#).
- Making deep learning accessible <http://timdettmers.com/>