

Python Anaconda distribution, PyCharm IDE, Conda & PyPI Packages, and Theano Package: Installation Guides

IMPORTANT: Computer Hardware and Operating System Assumptions

In the following installation guides, we assume that your machine runs on either Mac or Windows.

If you happen to use Linux or another operating system, certain setups can get hairier and there may be a need to consult Dr. Google heavily. You may contact the Teaching Assistants for individualized assistance, although there is no definite promise we can solve non-Mac, non-Windows problems.

1. Python Installation

We recommend [Continuum Analytics'](#) Anaconda distribution of Python v2.7 and compatible popular scientific computation packages.

Download the graphical installer appropriate for your machine's processor (32-bit / 64-bit) and operating system (Mac / Windows) [here](#). Run the installer and follow its instructions to install the software into a folder you prefer.

Note that the Anaconda installation folder path MUST NOT CONTAIN BLANK SPACES, because spaces occasionally cause bugs in dependent open-source software packages.

1.1. Anaconda Academic License

Continuum offers free Anaconda advanced computation optimization add-ons for academic use. You may obtain an Academic License [here](#) and follow the instructions in the company's email after you submit your request.

Among all things, this Academic License gives you a free upgrade of the default BLAS (Basic Linear Algebra Subprograms) libraries of Anaconda Python to Intel's Math Kernel Library (MKL), which is among the fastest BLAS'es around. This enables faster math, saving time for you and your machine.

The commands to install MKL are already included in the pre-included installation script file discussed in *Section 3. Additional Conda & PyPI Packages Installation*.

2. JetBrains PyCharm IDE Installation [*Recommended but Optional*]

[JetBrains](#), a vendor of popular integrated development environments (IDEs) for professional software developers, has an excellent IDE named PyCharm for Python.

We recommend PyCharm, but you may feel free to go with any other Python IDE(s) of your own choice.

If you do opt for PyCharm, go [here](#) to download its free Community Edition and install it onto your machine.

Note that you typically need to manually configure your installed IDE (PyCharm or other) to link it to the a Python backend ("Python interpreter"). There should be easy-to-follow online guides on how to do this for each IDE.

3. Additional Conda & PyPI Packages Installation

After

4. Theano Package Installation, Configuration & Tutorial

[Theano](#) is a Python library that enables using a compatible GPU (Graphical Processing Unit) of the computer for numerical computation, which is far superior in performance terms to computation by the computer's CPU (Central Processing Unit). **Currently, the most, if not only, effectively Theano-supported GPUs are those by NVIDIA.**

Because of its huge performance-boosting benefits, Theano is prominent in extremely data-intensive, large-scale Machine Learning applications such as those in cutting-edge Deep Learning.

Having said that, Theano is still very young (read: buggy), and its setup, configuration and usage demand a high degree of risk-taking and perseverance.

We will use Theano only on a **very limited, experimental basis**, and will install it in such a way that if your computer does not have an NVIDIA GPU, Theano will fall back to NumPy, which is Python's default numerical computation library.

4.1. Theano Installation and Configuration on Mac

You may refer to the Theano installation and configuration guide for Mac [here](#).

The key steps are:

- i. Install the **Anaconda Python v2.7 distribution** (this should be already done, per instructions in Section 1. *Python Installation*)
- ii. Install the **Clang compiler** through installing the XCode app from the Apple App Store and running it once to install the command-line tools
- iii. Install the **CUDA graphics driver and toolkit** from NVIDIA in order to use the GPU to perform numerical computation
- iv. Update your *.bash_profile* file with the following:

```
export PATH="/path/to/your/cuda/root:$PATH"
export CUDA_ROOT="/path/to/your/cuda/root"
export LD_LIBRARY_PATH="/path/to/your/cuda/lib"
export THEANO_FLAGS="cuda.root=/path/to/your/cuda/root,device=gpu,force__device=False,
floatX=float32,blas.ldflags=-L</your/anaconda/lib/> -<BLAS flag 1> -<BLAS flag 2> -<BLAS flag 3>"
```

For Mac, *</path/to/your/cuda/root>* is usually */usr/local/cuda/bin*, and *</path/to/your/cuda/lib>* is usually */usr/local/cuda/lib*.

</your/anaconda/lib/> is the path to the *lib* subfolder in your Anaconda installation folder.

One common source of error in setting up Theano on Mac concerns the BLAS (Basic Linear Algebra Subprograms) libraries. Refer [here](#) for some troubleshooting guidance on this issue.

4.2. Theano Installation and Configuration on Windows

You may refer to the Theano installation and configuration guide for Mac [here](#).

The key steps are:

- i. Install the **Anaconda Python v2.7 distribution** (this should be already done, per instructions in Section 1. *Python Installation*)
- ii. Install **Visual Studio Express 2010 (C++ version)**
- iii. Install the **Windows Software Development Kit v7.1**
- iv. Install the **CUDA graphics driver and toolkit v5.5** from NVIDIA in order to use the GPU to perform numerical computation
- v. Install the **Microsoft Visual C++ Compiler for Python v2.7**
- vi. Install **GCC**

4.3. Theano Configurations *[for info only]*

Theano has a large number of configuration levers, the most common of which are discussed [here](#). You do not need to pay attention to those for this course.

4.4. Theano Tutorials *[for info only]*

DeepLearning.net offers a comprehensive series of tutorials on Theano [here](#). Again, these are not required for this course. You can look at these at your own time if you are keen.