Setting Up Python Environment

# Overview

These are notes on how to set up a Python Desktop environment to perform Deep Learning, host a small server and maintain a small database.

# Choice of Python

To keep decision making as simple as possible this installation guide is for Python 3.

Check current Python version installed

$ python --version

Check all python version available

$ ls -ls /usr/bin/python\*

You should see references to Python version 2.7 and 3.6 or above.

# Installing PIP

Pip is a package management system that allows for easy installation of software libraries. PIP stands for "Pip Installs Packages" which is a recursive acronym. Because we want to use Python 3 we need to ensure that Pip3 is used.

To check you have the correct version perform the following command:

$ pip3 --version

pip 9.0.1 from /usr/lib/python3/dist-packages (python 3.6)

To upgrade pip3 use the following command:

$ pip3 install --upgrade pip

To install and uninstall packages you can simply use the following commands:

$ pip3 install some-package-name

$ pip3 uninstall some-package-name

# Installing and using a virtual environment

It is recommended to create a virtual environment which are isolated from each. That way each project can have a set of libraries that won’t cause conflicts. To set up a virtual environment you need to install virtualenv with pip3:

$ pip3 install --user --upgrade virtualenv

Now it is possible to set up an isolated virtual environment. Change directory to the folder which will be your new project and execute the following command:

$ virtualenv env

Every time you want to use this virtual environment you need to use the source command in the project directory you want to work

$ source /path/to/project/bin/activate

While the environment is active any packages installed with pip3 will installed in this isolated environment.

It is also possible to a virtual environment with Pycharm, you have to be careful in your choice of whether the environment will be available to multiple projects.

# Installing Pycharm on Linux Mint

# Installing packages to perform Deep Learning

To perform basic deep learning, the following packages are recommended

$ pip3 install --upgrade jupyter matplotlib numpy pandas scipy scikit-learn

To check if everything went okay execute the following command, there should be no output

$ python3 -c ‘import jupyter, matplotlib, numpy, pandas, scipy, sklearn’

To start up jupyter execute the following command

$ jupyter notebook

To work with TensorFlow additional packages are necessary. Choose package name tensorflow-gpu if you want to use your graphical processor.

$ pip3 install --upgrade tensorflow

$ python3 -c 'import tensorflow as tf; print(tf.\_\_version\_\_)'

To work with Keras additional packages are necessary.

$ pip3 install --upgrade pillow h5py keras

I suggest at this point you use the ‘Hello, world!’ implementation of Keras, with MNIST handwriting, to test your environment.

To work with Flask and Django install the following packages