Practical Python For Tensorflow

Tony Xu

9/16/2017

Software Requirements

- OS : Ubuntu 16.04 (14.04)
- Python 2.7 (3.x)
- OpenCV 3.x
- TensorFlow
- Many supported packages

Download and Install OpenCV and Tensorflow

- http://www.pyimagesearch.com/2015/06/22/install-opencv-3-0-and-python-2-7-on-ubuntu/
- https://www.tensorflow.org/install/install_linux

Python

What is Python

- **A very-high-level language – human readable**
- **Simple to use, but it is a real powerful programming language.**
- **An interpreted language**
  - No compilation and linking (vs. c/c++)
  - Easy to debug (using lots of print commands)

What is Python (Continued)

- **Python uses many packages (modules/libs)**
  - Pyside/PyQt for GUI
  - Numpy for numerical operation
  - **OpenCV for computer vision**
  - **Tensorflow for machine learning**
  - Many others
  - You can build your own - extensible
- **Used in C , or use C in python**
- **Mainly use for prototyping**
  - Performance issues
  - Source code leaking issues

Python Releases

- **Python 2 (2.7) vs. Python 3 (3.6)**
  - No backwards compatibility
  - Porting tools are available, but incomplete
  - We will learn Python 2.7
- **Linux/Windows/Mac OS**
  - Please check python versions on your system

Determine Your Python

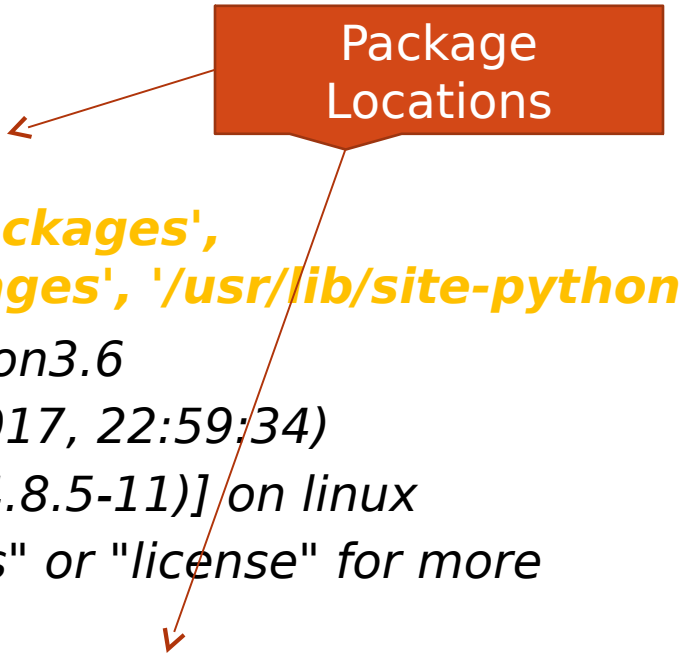
- **We are using Ubuntu 16.04 (or 14.04)**
- **If you are using Windows:**
  - Make a dual boot system (not easy)
  - Install virtualbox
    - https://www.virtualbox.org/wiki/Downloads
    - If you GPU (Nvidia) you may want to use VMWare virtualization platform.
- **You need Python version 2.7 (preferred) or 3.5 (not 3.6) for Tensorflow**
- **Check your Python version on your system.**

Install Python on Ubuntu

- **Get all needed libraries:**
  - sudo apt-get install build-essential checkinstall
  - sudo apt-get install libreadline-gplv2-dev libncursesw5-dev libssl-dev libsqlite3-dev tk-dev libgdbm-dev libc6-dev libbz2-dev
- **Download version:**
  - version=2.7.13
  - cd ~/Downloads/
  - wget https://www.python.org/ftp/python/$version/Python-$version.tgz
- **Extract and go to the directory:**
  - tar -xvf Python-$version.tgz
  - cd Python-$version
  - ./configure make
  - sudo checkinstall

Find Out What You Have

- *[fpga@localhost pythonenv]$ python*
  - ***Python 2.7.5** (default, Jun 17 2014, 18:11:42)*
  - *[GCC 4.8.2 20140120 (Red Hat 4.8.2-16)] on linux2*
  - *Type "help", "copyright", "credits" or "license" for more information.*
  - *>>> import site*
  - *>>> site.getsitepackages()*
  - ***['/usr/lib64/python2.7/site-packages', '/usr/lib/python2.7/site-packages', '/usr/lib/site-python']***
- *[fpga@localhost pythonenv]$ python3.6*
  - ***Python 3.6.2 (**default, Jul 18 2017, 22:59:34)*
  - *[GCC 4.8.5 20150623 (Red Hat 4.8.5-11)] on linux*
  - *Type "help", "copyright", "credits" or "license" for more information.*
  - *>>> import site*
  - *>>> site.getsitepackages()*
  - ***['/usr/lib64/python3.6/site-packages', '/usr/lib/python3.6/site-packages']***



A Big Problem:
Not Backwards Compatible

- **Issues occur:**
  - When many versions of Pythons coexisting on the system: e.g. 2.7 vs.3.6.
  - When applications use different versions of same packages which are **NOT** backward compatible
- **Solutions:**
  - Virtual environment

Setup Virtual Environment

- **A tool: virtualenv**
- **Create a Separate Env.**
  - $ [sudo] pip install virtualenv
  - virtualenv -p python p27env
- **Use it**
  - $ source ./p27env/bin/activate
  - Using python here…(installing packages)
  - $ deactivate

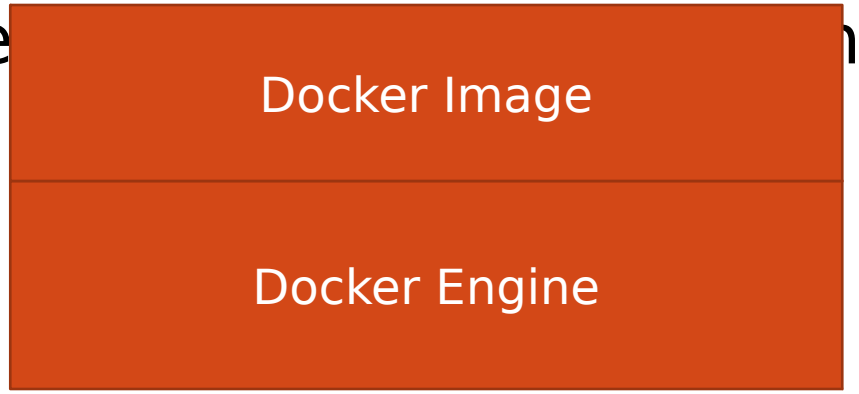
A Minute On Docker

- **A better alternative which is getting popular in industries recently**
- **https://www.docker.com/**
- **Wikipedia defines Docker as**

  an open-source project that automates the deployment of software applications inside **containers** by providing an additional layer of abstraction and automation of **OS-level virtualization** on Linux

Docker Images

- Docker Images share between developers/users
- Docker images include everything: OS, libraries, applications
- Hosts nee ne



Best Python IDE

- **Pycharm (highly recommended)**
  - https://www.jetbrains.com/pycharm/
- **For Others**
  - http://stackoverflow.com/questions/81584

Start Python

- **Interactive mode:**
  - Go to your virtual environment
  - Type in “python” or “python3.5” in command line
- **Script mode:**
  - Go to your virtual environment
  - Use your IDE to program
    - Pycharm e.g.
  - File with .py extension
  - Execution: python mycode.py
  - Execution without typing python:
    - Add this first line #! /usr/bin/env python

Print is Your Best Friend

- **Help you learn and debug, use it whenever you have doubts**
- **Python 3.x**
  - >>> print(‘abc’)
- **Python 2.7 both legal**
  - >>> print(‘abc’)
  - >>> print ‘abc’
- **Print out floats (You need that in your Tensorflow)**
  - >>> var1 = 1.0
  - >>> val2 = 2.2
  - >>> print(“%.2f kg = %.2f lb”) % (var1, var2)
  - >>> print((“%.2f kg = %.2f lb”) % (var1, var2)) (python 3.x)
- **Help(variable) is also useful**

Something You Need to Know

- **Convert strings to numbers**
  - >>> a = "545.2222"
  - >>> float(a)
  - 545.22220000000004
  - >>> int(float(a))
  - 545
- **Priority : always use (parentheses) as a good habit!**
  - >>> 2 + 3 * 5
  - >>> (2+3) * 5
- **Make sure types:**
  1. >>> 2/5 #what do you think you will get?
  2. >>> 2%5
  3. >>> 2 .0/5

“\” A Special Character

- **\n -> newline**
- **\t -> tab**
- **\\ -> backslash**
- **Trouble to open your files:**

```
>>> filename = "M:\nickel_project\reactive.smi" # DANGER!
>>> filename = "M:\\nickel_project\\reactive.smi" # Better!
>>> filename = "M:/nickel_project/reactive.smi" # Usually works
>>> filename = r"M:\nickel_project\reactive.smi"
```

Good Coding Habits

- **KISS : Keep it Simple, Stupid**
- **Always put comments in your code**
  - # for single line
  - “”” multiple lines “””
- **Don’t change variable type**
  - X = 1 …. X = “abc”
- **Meaningful function/variable names**
  - Variable name: Something; SecondPerMinute
  - Function name: doSomething(); displayWindow()
- **Unit Test**

Code Study 1

- """The os and sys modules provide numerous tools to deal with filenames, paths, directories."""
- **import os, sys**
- """OpenCV Computer Vision package which includes numerous functions for image related processing """
- **import cv2**

For Loop in Python

- **for <variable> in <sequence>:**

  <statements>

- >>> languages = ["C", "C++", "Perl", "Python"]

  >>> for x in languages:

  >>> print(x)

Condition Flow Control

- **if <evaluation> :**
  block of code
  else :
  block of code
- >>> if x < 0:
- >>> x = 0
- >>> print('Negative changed to zero')
- >>> else :
- >>> print (‘positive number’)

Concise Version Is Hard

- >>> d =[]
- >>> a = [2,3,4,5,6,7,8,9,0]
- >>> xyz = [0,12,4,6,242,7,9]
- >>> for x in xyz:
- >>> if x in a:
- >>> d.append(x)
- **# it can be do this way**:
- >>> **d = [x for x in xyz if x in a]**

Code Study 2

- **# assign “Training” to data_dir as a string type**
- **data_dir = "Training" *#path for the folder***
- ***#list all the directories which hold all different images in each directory***
- **directories = [d for d in os.listdir(data_dir)**
  if os.path.isdir(os.path.join(data_dir, d))]
- “””os.listdir () :The method **listdir()** returns a list containing the names of the entries in the directory given by path. The list is in arbitrary order. It does not include the special entries '.' and '..' even if they are present in the directory”””
- #os.path.isdir() : determine if current path is a directory
- #os.path.join(x,y): give you x\y in Windows, x/y in Linux
- #it is different than join function in Python

Code Study 3

- **for d in directories:**

```
    images = []
    label_dir = os.path.join(data_dir, d)
    file_names = [os.path.join(label_dir, f)
                  for f in os.listdir(label_dir)
                  if f.endswith(".ppm") or
f.endswith(".JPG")
                  or f.endswith(".png")]
```

- **# can you tell me what this code is doing?**

More Human Readable Version

- file_names = [os.path.join(label_dir, f)
  for f in os.listdir(label_dir)
  if f.endswith(".ppm") or f.endswith(".JPG")
  or f.endswith(".png")]
- #But we want you do this way:
- #KISS: Keep It Simple and Stupid
- file_names = []
- for f in os.listdir(label_dir)
- if f.endswith(".ppm") or f.endswith(".JPG“) or f.endswith(".png"):
- file_names.append(os.path.join(label_dir, f))

Code Study 4

- **for** f **in** file_names:
  img = cv2.imread(f)
  if img **is not** None:
  print("**Processing**" + f)
- **# img is a MAT class, please refer to OpenCV document**
- **# img has many members and functions**
- **# includes img.shape[1] image width**

Code Study 5

- M = cv2.getRotationMatrix2D((img.shape[1] / 2, img.shape[0] / 2),10, 1)
- ***# rotation matrix by 10 degrees***
- rotate1 = cv2.warpAffine(img, M, (img.shape[1], img.shape[0]))
- ***# rotate image and assign it back***
- blur1 = cv2.GaussianBlur(img, (5, 5), 3)
- ***# 5 by 5 kernel, sigma 3***

Code Study 6

- cv2.imwrite(f[0:-4] + **"_rotate1.jpg"**, rotate1)
- # write to image to a file
- # with name f[0:-4] cutting of last 4 characters

Homework

- Write a python program to display all your directories and files