Machine Learning with Python

R.Jayabhaduri & Aditi Prakash

Sri Venkateswara College of Engineering

November 12,2018

Outline

- **Anaconda**
- **Conda**
- **Keras**
- **TensorFlow**
- **PyTorch**
- **Librosa**

Introduction to Anaconda

Anaconda is a Python distribution that is particularly popular for data analysis and scientific computing

- Open source project
- Available for Windows, Mac OS X and Linux
- Includes many popular packages: NumPy, SciPy, Matplotlib, Pandas,IPython, Cython
- Includes Spyder, a Python development environment
- Includes conda, a platform-independent package manager

Introduction to Conda

Simplifies installation of Python packages

- Platform-independent package manager
- An environment management system that runs on Windows, Mac OS and Linux
- Provides "virtual environment" capabilities

To Update Conda : conda update −−all

Instructions for setting-up Deep Learning Environment

1. Login to Ubuntu : Open Terminal
2. Check the availability of Python (default) in Ubuntu : $ which python
3. Download Anaconda-3 from Internet and copy to /home
4. Open terminal, install Anaconda-3
   $ bash Anaconda3-5.0.1-Linux-x86_64.sh
5. By default, Anaconda will be installed at home/root. Press Enter to confirm
6. Open new terminal to activate conda
7. To list the packages installed by conda
   $ conda list
8. To check the Python environment : which python
   /root/anaconda3/bin/python

Python – Using terminal

Type python in terminal and test

$python
>>> print ("Welcome to SVCE")

Python Test – Using iPython

```
$ipython
In [1]: print ("Welcome to STTP MLDLTISP'18")
Welcome to STTP MLDLTISP'18
In [2]:
```

Installing Python packages with Conda

Library	Command
Keras	conda install keras
TensorFlow	conda install tensorflow
OpenCV	conda install opencv
Librosa	conda install -c conda-forge librosa
PyTorch	conda install pytorch torchvision -c pytorch

Table 1: Python packages with Conda

Introduction to Keras

What is Keras

- Built on top of TensorFlow, CNTK, or Theano.
- Supports both convolutional networks and recurrent networks, as well as combinations of the two.
- Runs seamlessly on CPU and GPU

Why Keras

- Allows for easy and fast prototyping (through user friendliness, modularity, and extensibility)

How to install keras

Introduction to TensorFlow

What is TensorFlow

- Open source software library for numerical computation using data flow graphs
- Originally developed to conduct machine learning and deep neural networks research

Why TensorFlow

- Python API
- Portability: deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API

TensorFlow

Getting Started to TensorFlow

```
import tensorflow as tf
```

How to install TensorFlow

```
$ conda install tensorflow
```

What's a tensor?

An n-dimensional array

- 0-d tensor: scalar (number)
- 1-d tensor: vector
- 2-d tensor: matrix

PyTorch

Introduction to PyTorch

- open source machine learning library for Python, based on Torch
- Used for applications such as Natural Language Processing

How to install PyTorch

$ conda install pytorch torchvision -c pytorch

PyTorch

Check Installation of PyTorch

Open terminal and type python3

```
import torch
a=torch.tensor([[1,2],[3,4]])
print (a)
```

Output

```
tensor([[1, 2],[3, 4]])
```

Librosa

Introduction to Librosa

- A python package for music and audio analysis
- Provides the building blocks necessary to create music information retrieval systems

How to install Librosa

- Pypi (Simplest way - through Python Package Index (PyPI))
  $ **pip install librosa**
- Conda/Anaconda environments
  $ **conda install -c conda-forge librosa**

OpenCV

Introduction to OpenCV

Open Source Computer Vision Library

How to install OpenCV

$ conda install opencv

OpenCV

Introduction to OpenCV

Open Source Computer Vision Library

How to install OpenCV

$ conda install opencv

NumPy

Introduction to NumPy

The fundamental package for scientific computing with Python

Getting started

import numpy as np

NumPy

Introduction to NumPy - Arrays

```
a = np.array([1, 2, 3]) # Create a rank 1 array
print(type(a))
print(a.shape)
print(a[0], a[1], a[2])
a[0] = 5
print(a)
```