Assignment-1

Study of Deep Learning Packages: Tensorflow,Keras,Theano and PyTorch.Document the distinct features and functionality of the packages.

Aim : Study of following Deep learning Packages and their installations :

- Tensor Flow
- Keras
- Theano
- PyTorch

Theory :

1. What is Deep learning ?
2. What are various packages in python for supporting Machine Learning libraries and which are mainly used for Deep Learning ?
3. Compare Tensorflow / Keras/Theno and PyTorch on following points(make a table) :
    a. Available Functionality
    b. GUI status
    c. Versions.
    d. Features
    e. Compatibilty with other enviornments.
    f. Specific Applictaion domains.
4. Enlist the Models Datasets and pretrained models, Libraaries and Extensions , Tools related to Tensorflow also discuss any two casestudies like (PayPal, Intel, Etc.) related to Tensor Flow. [Ref:https://www.tensorflow.org/about]
5. Explain simple Theano program.
6. Explain PyTorch Tensors . And also explain Uber’ s Pyro, Tesla Autopilot.[https://pytorch.org/]
7. Explain the Keras Ecosystem.(kerastuner,kerasNLP,kerasCV,Autokeras and Modeloptimization.) Also explain following concepts related to keras : 1. Developing sequential Model 2. Training and validation using the inbuilt functions 3. Parameter Optimization. [Ref: https://keras.io/]

Steps/Algorithm :

Installation of TensorFlow on Ubuntu :

Prerequisites for Installing TensorFlow on Ubuntu :

- An Ubuntu Linux system (16.04 version or later)
- Python 3.5 or the latest version
- Pip 19.0 or newer versions
- A user account with sudo privileges

1. Install the Python Development Environment

- You need to download Python, the PIP package, and a virtual environment. If these packages are already installed, you can skip this step.
- You can download and install what is needed by visiting the following links:
- https://www.python.org/
- https://pip.pypa.io/en/stable/installing/
- https://docs.python.org/3/library/venv.html
- To install these packages, run the following commands in the terminal:
  a) sudo apt update
  b) sudo apt install python3-dev python3-pip python3-venv

2. Create a Virtual Environment

Navigate to the directory where you want to store your Python 3.0 virtual environment. It can be in your home directory, or any other directory where your user can read and write permissions.

1. mkdir tensorflow_files

2. cd tensorflow_files

Now, you are inside the directory. Run the following command to create a virtual environment:

3. python3 -m venv virtualenv

The command above creates a directory named virtualenv. It contains a copy of the Python binary, the PIP package manager, the standard Python library, and other supporting files.

3. Activate the Virtual Environment

- source virtualenv/bin/activate

Once the environment is activated, the virtual environment's bin directory will be added to the beginning of the $PATH variable. Your shell's prompt will alter, and it will show the name of the virtual environment you are currently using, i.e. virtualenv.

4. Update PIP

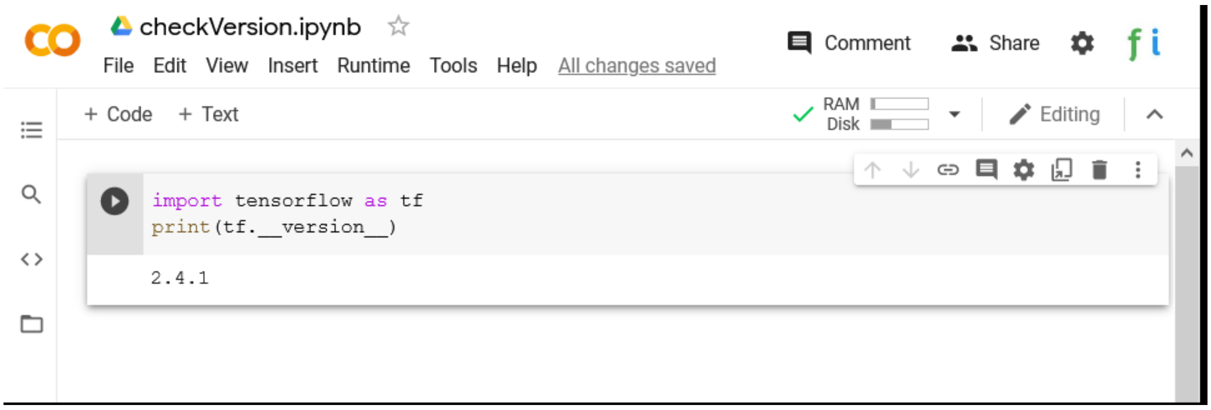
- pip install --upgrade pip

5. Install TensorFlow

The virtual environment is activated, and it's up and running. Now, it's time to install the TensorFlow package.

- pip install -- upgrade TensorFlow

To check if TensorFlow has been installed successfully, run the following lines of code on Jupyter Notebook. Print the version of TensorFlow, and perform a mathematical operation.



```
import tensorflow as tf
print(tf.__version__)
```

```
2.4.1
```

The lines of code above can be run in command prompt or on any Python IDE. If someone is using Jupyter Notebook for all of their code, for instance, they can run the same commands here as well.

Installation of Keras on Ubuntu :

Prerequisites

- A Linux machine with access to a command-line/terminal
- The Python 3.5 – 3.8 development environment

STEP 1: Install and Update Python3 and Pip

Skip this step if you already have Python3 and Pip on your machine.

If not, open the terminal and enter the following command, depending on your Linux distribution:

CentOS / RedHat:

```
sudo yum install python3 python3-pip
```

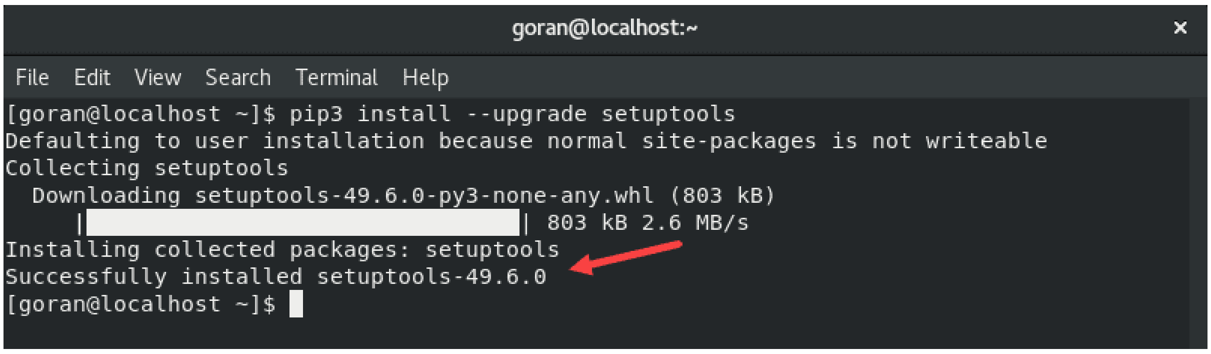
Then, run this command to upgrade Pip:

```
sudo pip3 install ––upgrade pip
```

STEP 2: Upgrade Setuptools

To upgrade **setuptools**, enter the following:

```
pip3 install ––upgrade setuptools
```



```
[goran@localhost ~]$ pip3 install --upgrade setuptools
Defaulting to user installation because normal site-packages is not writeable
Collecting setuptools
  Downloading setuptools-49.6.0-py3-none-any.whl (803 kB)
     |████████████████████████████████| 803 kB 2.6 MB/s
Installing collected packages: setuptools
Successfully installed setuptools-49.6.0
[goran@localhost ~]$ 
```

Without this step, you may receive errors about certain packages requiring a different **setuptools** version than the one you have on your system.

STEP 3: Install TensorFlow

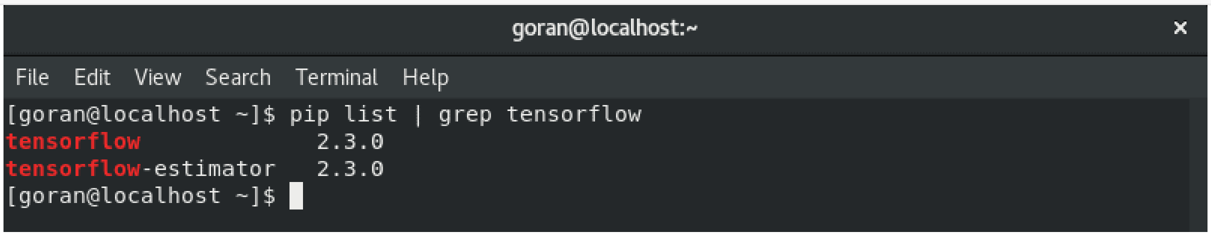
The TensorFlow installation is straightforward. Use Pip and this command to install it::

```
pip3 install tensorflow
```

Let the download and installation finish.

Verify the installation was successful by checking the software package information:

```
pip3 show tensorflow
```



```
[goran@localhost ~]$ pip list | grep tensorflow
tensorflow             2.3.0
tensorflow-estimator   2.3.0
[goran@localhost ~]$ 
```

STEP 4: Install Keras

Finally, install Keras with the following command:

```
pip3 install keras
```

Verify the installation by displaying the package information:

```
pip3 show keras
```

STEP 5: Install Keras from Git Clone (Optional)

If you have Git on your system, you can use it to clone a copy of the Keras software package from GitHub.

To clone the Keras package from GitHub, enter the following:

```
git clone https://github.com/keras-team/keras.git
```

Once the download completes, switch to the **/keras** directory:

```
cd keras
```

From there, run the **Keras** python installer:

Installation of Theano on Ubuntu :

To install the Theano package in Linux we have to follow the following steps:

Step 1: First of all, we will install Python3 on our Linux Machine. Use the following command in the terminal to install Python3.

```
sudo apt-get install python3
```

Step 2: Now, install the pip module which is required to install the packages in Python3. So we use the following command for installation:

```
sudo apt install python3-pip
```

Step 3: Now, install the Theano package by using pip module. Use the following command to install the Theano package.

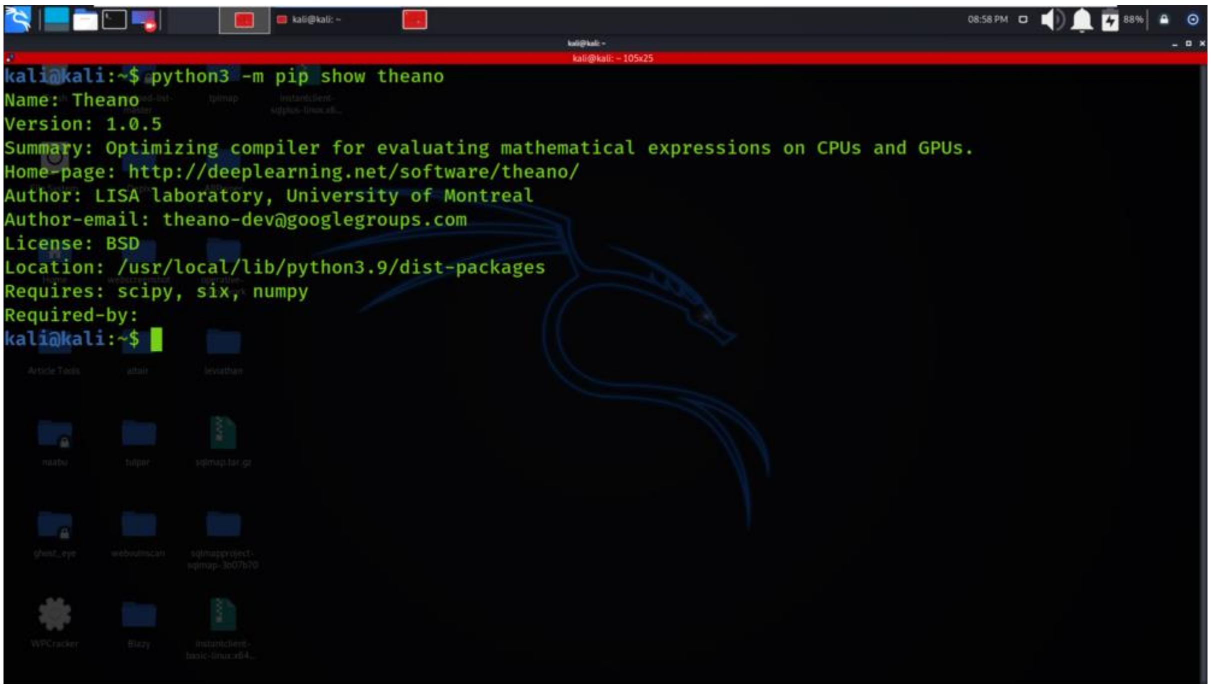
```
sudo pip3 install theano
```

Verifying Theano package Installation on Linux using PIP

To verify if the Theano package has been successfully installed in your system run the below command in Terminal:

```
Python3 -m pip show theano
```

You'll get the below message if the installation is complete:



```
kali@kali:~$ python3 -m pip show theano
Name: Theano
Version: 1.0.5
Summary: Optimizing compiler for evaluating mathematical expressions on CPUs and GPUs.
Home-page: http://deeplearning.net/software/theano/
Author: LISA laboratory, University of Montreal
Author-email: theano-dev@googlegroups.com
License: BSD
Location: /usr/local/lib/python3.9/dist-packages
Requires: scipy, six, numpy
Required-by:
kali@kali:~$
```

Installation of PyTorch on Ubuntu :

First, check if you are using python's latest version or not.

Because PyGame requires python 3.7 or a higher version, make sure you are using the latest version of python. Type the below command to check the version of python.

```
python3 –version
```

Note: If Python is not installed, refer to install python in Linux.

Check if you are using the latest version of pip or not:

The pip is a python package installer, if you want to use any external package in your python file you first install it in your local system using pip. Run the following command to check the version of pip.

```
pip3 –version
```

If you are already using the new pip version so follow the below steps if not, so refer to the article on how to install pip in Linux.

Switch to root user and update Linux packages if you are not using the latest pip version:

Open the terminal and make sure you are the root user. Run the following command to switch to root user.

```
sudo su
```

Update Linux packages using the below command.

```
apt update
```

After this step, you are ready to install PyTorch on your Linux system.

Installing PyTorch

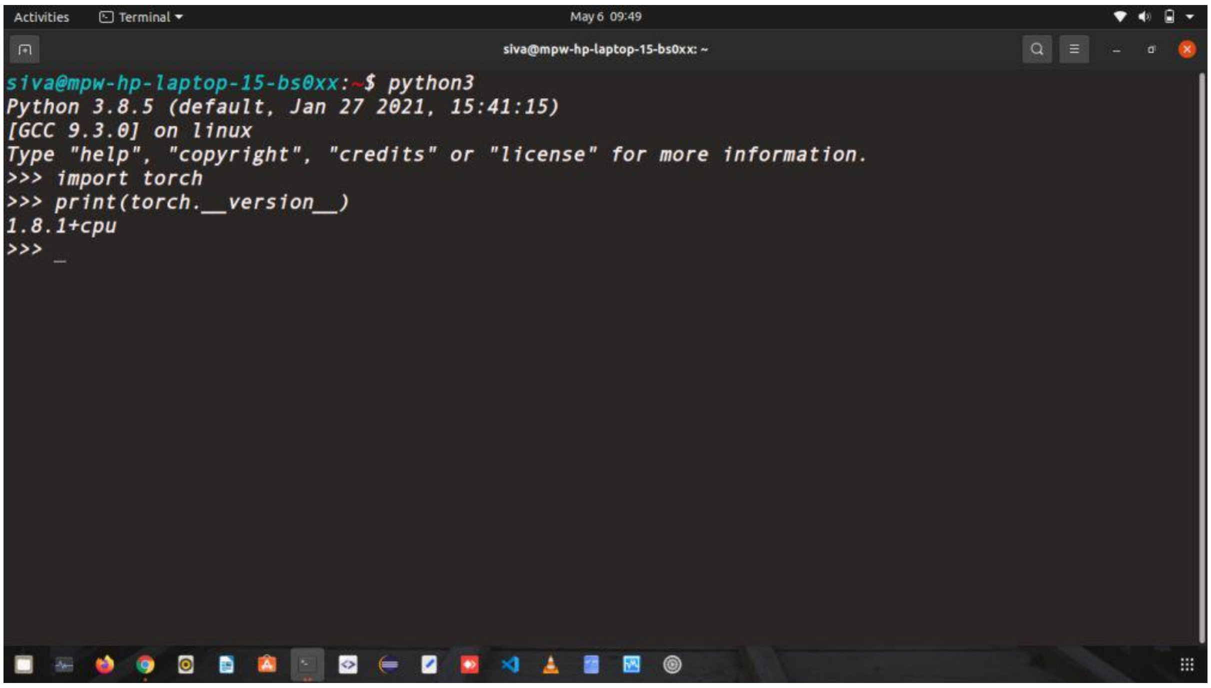
```
pip3 install torch==1.8.1+cpu torchvision==0.9.1+cpu torchaudio==0.8.1 -f
https://download.pytorch.org/whl/torch_stable.html
```

The above command was used to install PyTorch in the system that didn't have GPU. You just copy the command and paste it into the terminal and run it.

The below command is used to install PyTorch on a system which has GPU. Make sure you have python 3.7 or higher.

```
pip3 install torch torchvision torchaudio
```

To make sure PyTorch is installed in your system, just type python3 in your terminal and run it. After that type import torch for use PyTorch library at last type and run print(torch.__version__) it shows which version of PyTorch was installed on your system if PyTorch was installed on your system.



```
siva@mpw-hp-laptop-15-bs0xx:~$ python3
Python 3.8.5 (default, Jan 27 2021, 15:41:15)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch
>>> print(torch.__version__)
1.8.1+cpu
>>> _
```

Python Libraries and functions required :

The Python Standard Library contains the exact syntax, semantics, and tokens of Python. It contains built-in modules that provide access to basic system functionality like I/O and some other core modules. Most of the Python Libraries are written in the C programming language. The Python standard library consists of more than 200 core modules. All these work together to make Python a high-level programming language. Python Standard Library plays a very important role. Without it, the programmers can't have access to the functionalities of Python.Let's have a look at some of the commonly used libraries:

1. **TensorFlow:** This library was developed by Google in collaboration with the Brain Team. It is an open-source library used for high-level computations. It is also used in machine learning and deep learning algorithms. It contains a large number of tensor operations.

```
import tensorflow as tf
```

2. **Matplotlib:** This library is responsible for plotting numerical data. And that's why it is used in data analysis. It is also an open-source library and plots high-defined figures like pie charts, histograms, scatterplots, graphs, etc.

```
import matplotlib
matplotlib.__version__
```

3. **Pandas:** Pandas are an open-source machine learning library that provides flexible high-level data structures and a variety of analysis tools.. Pandas support operations like Sorting, Re-indexing, Iteration, Concatenation, Conversion of data, Visualizations, Aggregations, etc.

```
import pandas as pb
```

4. **Numpy:** The name “Numpy” stands for “Numerical Python”. It is the commonly used library. It is a popular machine learning library that supports large matrices and multi-dimensional data. It consists of in-built mathematical functions for easy computations.

```
import numpy as np
```

5. **SciPy:** The name “SciPy” stands for “Scientific Python”. It is an open-source library used for high-level scientific computations. This library is built over an extension of Numpy. It works with Numpy to handle complex computations. While Numpy allows sorting and indexing of array data, the numerical data code is stored in SciPy. It is also widely used by application developers and engineers.

```
sudo apt-get install python-scipy python-numpy
```

6. **sklearn** : Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib. For importing train_test_ split use

```
from sklearn.model_selection import train_test_split
```

Sample Code with Comments :

I. Tensorflow Test program :

Code :

```
# importing tensorflow
import tensorflow as tf

# creating nodes in computation graph
node1 = tf.constant(3, dtype=tf.int32)
node2 = tf.constant(5, dtype=tf.int32)
node3 = tf.add(node1, node2)

# create tensorflow session object
sess = tf.Session()

# evaluating node3 and printing the result
print("Sum of node1 and node2 is:",sess.run(node3))

# closing the session
sess.close()
```

Output :

```
Sum of node1 and node2 is : 8
```

II. Theano Test Program :

Code :

```
# Python program showing
# addition of two scalars

# Addition of two scalars
import numpy
import theano.tensor as T
from theano import function

# Declaring two variables
x = T.dscalar('x')
y = T.dscalar('y')

# Summing up the two numbers
z = x + y

# Converting it to a callable object
# so that it takes matrix as parameters
f = function([x, y], z)
f(5, 7)
```

Output :

```
array(12,0)
```

III. PyTorch Test Program :

Code :

```
import torch.nn as nn

# flatten the tensor into
class Flatten(nn.Module):
   def forward(self, input):
       return input.view(input.size(0), -1)

#sequential based model
seq_model = nn.Sequential(
           nn.Conv2d(1, 10, kernel_size=5),
           nn.MaxPool2d(2),
           nn.ReLU(),
           nn.Dropout2d(),
           nn.Conv2d(10, 20, kernel_size=5),
           nn.MaxPool2d(2),
           nn.ReLU(),
           Flatten(),
           nn.Linear(320, 50),
           nn.ReLU(),
           nn.Linear(50, 10),
           nn.Softmax(),
         )

net = seq_model
print(net)

array(12,0)
```

Output :

```
Sequential(
  (0): Conv2d(1, 10, kernel_size=(5, 5), stride=(1, 1))
  (1): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode
=False)
  (2): ReLU()
  (3): Dropout2d(p=0.5)
  (4): Conv2d(10, 20, kernel_size=(5, 5), stride=(1, 1))
  (5): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode
=False)
  (6): ReLU()
  (7): Flatten()
  (8): Linear(in_features=320, out_features=50, bias=True)
  (9): ReLU()
  (10): Linear(in_features=50, out_features=10, bias=True)
  (11): Softmax()
)
```

IV. Keras Test Program :

Code :

```
# first neural network with keras tutorial
from numpy import loadtxt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
# load the dataset
dataset = loadtxt('pima-indians-diabetes.csv', delimiter=',')
# split into input (X) and output (y) variables
X = dataset[:,0:8]
y = dataset[:,8]
# define the keras model
model = Sequential()
model.add(Dense(12, input_shape=(8,), activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
# compile the keras model
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['ac
curacy'])
# fit the keras model on the dataset
model.fit(X, y, epochs=150, batch_size=10)
# evaluate the keras model
_, accuracy = model.evaluate(X, y)
print('Accuracy: %.2f' % (accuracy*100))
```

Output :

```
...
768/768 [==============================] - 0s 63us/step - loss: 0.4
817 - acc: 0.7708
Epoch 147/150
768/768 [==============================] - 0s 63us/step - loss: 0.4
764 - acc: 0.7747
Epoch 148/150
768/768 [==============================] - 0s 63us/step - loss: 0.4
737 - acc: 0.7682
Epoch 149/150
768/768 [==============================] - 0s 64us/step - loss: 0.4
730 - acc: 0.7747
Epoch 150/150
768/768 [==============================] - 0s 63us/step - loss: 0.4
754 - acc: 0.7799
768/768 [==============================] - 0s 38us/step
Accuracy: 76.56
```

Conclusion :

Tensorflow , PyTorch,Keras and Theano all these packages are installed and ready for Deep learning applications . As per application domain and dataset we can choose the appropriate package and build required type of Neural Network.