

# **Assignment No.1**

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

**Problem statement:** Study and installation of following Deep learning Packages:

- i. Tensor Flow
- ii. Keras
- iii. Theano
- iv. PyTorch

## **Software Requirements -**

1. Python (3.x recommended)
2. Jupyter Notebook

## **Prerequisites –**

- Basic understanding of Python programming
- Familiarity with the concepts of Neural Networks.

## **Theory:**

### **Installation of Tensorflow:**

#### **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>

**pip (Python Package Installer):** pip should be included automatically when you install Python. If not, you can install it manually by following the instructions on the [pip installation page](#).

#### **2. Create a Virtual Environment:(Optional but Recommended):**

Creating a virtual environment is recommended to keep your TensorFlow installation isolated from other Python projects. Run the following command:

`python -m venv tensorflow_env`

This will create a new virtual environment named `tensorflow_env` in the current directory.

### **Activate the virtual environment:**

`activate`

#### **3. Update PIP:**

`pip install --upgrade pip`

#### **4. Install TensorFlow:**

With your virtual environment activated, install TensorFlow using pip:

`pip install tensorflow`

#### **5. Verify Installation:**

After installation completes, you can verify TensorFlow installation by running Python in the activated virtual environment:

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

### **Installation of Keras on windows :**

Prerequisite : Python version 3.5 or above

1. Install and Update Python and Pip
2. Install TensorFlow:  
Verify the installation was successful by checking the software package information:  
`pip3 show tensorflow`
3. Verify TensorFlow Installation:  
Verify the installation was successful by checking the software package information:  
`pip3 show tensorflow`
4. Install Keras:  
`pip install keras`
5. Testing Keras Installation:  
`pip show keras`

### **Installation of Theano on windows 10:**

1. Install and Update Python and Pip
2. Install some requirement packages  
`conda install mingw libpython`  
`pip install numpy`  
`pip install scipy`
3. `pip install theano`
4. Testing Keras Installation:  
`pip show keras`

### **Installation of PyTorch :**

First, check if you are using python's latest version or not.Because PyGame requires python 3.7 or a higher version

1. Install and Update Python and Pip
2. `pip install torch==1.8.1+cpu torchvision==0.9.1+cpu torchaudio==0.8.1 -f https://download.pytorch.org/wheel/torch\_stable.html`
3. `pip show torch`

### **Python Libraries and functions required:**

#### **1. TensorFlow:**

Distinct Features:

TensorFlow is an open-source deep learning framework developed by Google.

It offers a comprehensive ecosystem for machine learning and deep learning.

TensorFlow provides both high-level APIs (like Keras) and low-level APIs for maximum flexibility.

It supports distributed computing and can run on various platforms, including CPU and GPU.

Functionality:

TensorFlow is known for its computational graph-based approach, making it highly efficient for deep learning.

It has extensive libraries for neural networks, including pre-built layers and optimizers.

TensorFlow also supports deployment on mobile and embedded devices, making it versatile.

#### **2. Keras:**

Distinct Features:

Keras is an open-source high-level neural networks API that can run on top of TensorFlow, Theano, or other deep learning frameworks.

It is designed for easy and rapid prototyping of neural networks.

Keras is user-friendly, with a simple and consistent API.

Functionality:

Keras is excellent for beginners and for quickly building and testing neural network models.

It provides a high-level interface for defining and training neural networks with minimal code.

Keras can seamlessly switch between different backend engines, such as TensorFlow and Theano.

### 3. Theano:

Distinct Features:

Theano is an open-source numerical computation library that specializes in optimizing mathematical expressions.

It is one of the early deep learning frameworks and has been influential in the field.

Theano focuses on CPU and GPU computation optimization.

Functionality:

Theano is used as a backend for other deep learning frameworks like Keras.

It is highly efficient for symbolic mathematics and can automatically optimize expressions for better performance.

While Theano has been influential, it is less actively developed now in favor of TensorFlow and PyTorch.

### 4. PyTorch:

Distinct Features:

PyTorch is an open-source deep learning framework developed by Facebook's AI Research lab.

It's known for its dynamic computation graph, which makes it more flexible for certain applications.

PyTorch has gained popularity for research and rapid experimentation.

Functionality:

PyTorch's dynamic computation graph allows for easy debugging and dynamic changes during runtime.

It is highly popular in the research community due to its flexibility and support for dynamic networks.

PyTorch also offers a wide range of libraries for natural language processing, computer vision, and reinforcement learning.

## Now let's understand some libraries of Python :

### 1. Pandas:-

One of the open-source Python libraries mainly used in data science and deep learning subjects is Pandas. The library provides data manipulation and analysis tools, which are used for analyzing data. The library relies on its powerful data structures for manipulating numerical tables and time series analysis.

The Pandas library offers a fast and efficient way to manage and explore data by providing Series and DataFrames, which represent data efficiently while also manipulating it in different ways.

To import pandas use

```
import pandas as pd
```

### 2. Numpy:-

One of the other well-known Python libraries, NumPy can be seamlessly utilized for large multi-dimensional array and matrix processing. It relies on a large set of high-level mathematical functions, which makes it especially useful for efficient fundamental scientific computations in deep learning.

NumPy arrays require a lot less storage area than other Python lists, and they are faster and more convenient to use. The data can be manipulated in the matrix, transposed, and reshaped with the library. NumPy is a great option to increase the performance of deep learning models without too much complex work required.

To import numpy use

```
import numpy as np
```

### 3. SCikit-learn:-

Scikit-Learn was originally a third-party extension to the SciPy library, but it is now a standalone Python library on Github. Scikit-Learn includes DBSCAN, gradient boosting, support vector machines, and random forests within the classification, regression, and clustering methods.

One of the greatest aspects of Scikit-Learn is that it's easily interoperable with other SciPy stacks. It is also user-friendly and consistent, making it easier to share and use data.

For importing train\_test\_split use

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

### **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.