

NAME: KSHITIJ VINOD SALI

CLASS: BE-A4, ROLL NO: 16av09

SUBJECT: LP-IV: DEEP LEARNING

LAB ASSIGNMENT - 01

Title: Comparative Analysis of Deep Learning Frameworks

Course Outcome: CO1 - Learn and use various Deep learning tools and packages

Date of Completion:

Assessment Grade/ Marks:

Assessor's Sign with Date:

Problem Statement: Study of deep learning packages: TensorFlow, Keras, Theano, and PyTorch. Document the distinct features and functionality of these packages.

Blooms Taxonomy Category: Understanding, Applying

Requirements: Ubuntu OS, Python 3/DE, Python packages and libraries.

Theory:

For installation of deep learning frameworks: TensorFlow, PyTorch, Keras and Theano. A critical step is using a Python virtual environment for TensorFlow, a best practice that

teaches essential dependency management and prevents software conflicts.

The installation procedure are as follows:

i) TensorFlow Installation on Ubuntu:

a) Install Python development environment - Update package list and install the necessary Python tools using terminal commands:

```
sudo apt update
```

```
sudo apt install python3-dev python3-pip  
python3-venv
```

b) Create a Virtual Environment - Navigate to a desired directory & create new directory for the project. Inside this directory, create virtual environment: `python3 -m venv virtualenv`

c) Activate Virtual Environment - Activate the newly created environment to ensure subsequent package installations are isolated: `source virtualenv/bin/activate`.

d) Update PIP: Upgrade the pip package manager to latest version within activate environment: `pip install --upgrade pip`

e) Install TensorFlow: Install the TensorFlow package using pip: `pip install tensorflow`

Note: TensorFlow needs a GPU or a multi-core CPU with CUDA support to run it efficiently.

ii) Keras Installation on Ubuntu:

a) Prerequisite Check - Ensure Python version 3.5 or higher is installed.

b) Install Python and Pip - If not already present, install Python 3 and pip: `Sudo apt install python3 python3-pip`

c) Upgrade Setup Tools and Pip - Ensure setup tools and pip: `Sudo apt pip3 install --upgrade pip pip3 install --upgrade setuptools`

d) Install TensorFlow Backend: Keras requires a backend engine. Install TensorFlow: `pip3 install TensorFlow`

e) Install Keras - Install keras package: `pip3 install keras`

iii) Theano Installation on Ubuntu:

a) Install Python 3 - Use terminal to install Python3: `Sudo apt-get install python3`

b) Install Pip - Install Python package manager for Python3: `Sudo apt install python3-pip`

c) Install Theano - Install Theano package using terminal: `python3 -m pip install theano`

- iv) PyTorch Installation on Ubuntu 20.04
- Version Check - Verify that recent version of Python3 and pip3 are installed.
 - Install PyTorch - Use the specific pip command provided, which points to a stable wheel repository to ensure compatibility. The command installs torch, torchvision and torchaudio: `pip3 install torch==1.8.1+cpu torchvision==0.9.1+cpu torchaudio==0.8.1+cpu`
- <https://download.pytorch.org/whl/torch-stable.html>

Algorithm:

- TensorFlow Test Program - This code imports the TensorFlow library, prints its version and executes a basic `tensor()` operation to confirm the runtime its functional.
- ```
import tensorflow as tf
print(tf.__version__)
Expected Output: 2.1.0 (or similar)
print(tf.reduce_sum(tf.random.normal(1)))
Expected Output: A tf.Tensor object with a scalar value.
```
- Keras Test Program - This test verifies the Keras API by importing it from TensorFlow and attempting to load a standard MNIST dataset, a common task in deep learning.
- ```
from tensorflow import keras
```

```

from keras import datasets
(train_images, train_labels), (test_images,
test_labels) = datasets.mnist.load_data()
# verify the package shape of loaded data
print (train_images.shape, test_images.shape)

```

iii) Theano Test Program: This program demonstrates Theano's core functionality: defining symbolic variables (x, y), building a computational graph ($z = x + y$) & compiling it into a callable Python function (f).

```

import numpy
import theano.tensor as T
x = T.dscalar('x')
y = T.dscalar('y')
z = x + y
f = functional([x, y], z)
# Expected Output: 12.0

```

iv) PyTorch Test Program: This simple test imports the torch library and prints its version to confirm the installation.

```

import torch
import torch.nn as nn
print (torch.__version__)

```

Inference: 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 for our requirement.

Some of the most popular packages are:

- TensorFlow
- Keras
- Theano
- Caffe
- PyTorch
- MXNet
- TensorRT
- TensorFlow.js
- TensorFlow Lite
- TensorFlow Hub
- TensorBoard
- TensorRT
- TensorFlow.js
- TensorFlow Lite
- TensorBoard

TensorFlow is an open source software library for numerical computation using data flow graphs. It was developed by Google Brain Team.

TensorFlow is used for various applications such as image processing, natural language processing, speech recognition, reinforcement learning, etc.

TensorFlow is a highly flexible and efficient framework for numerical computation, capable of running on multiple platforms (CPU/GPU).

TensorFlow is used for various applications such as image processing, natural language processing, speech recognition, reinforcement learning, etc.

TensorFlow is a highly flexible and efficient framework for numerical computation, capable of running on multiple platforms (CPU/GPU).

TensorFlow is a highly flexible and efficient framework for numerical computation, capable of running on multiple platforms (CPU/GPU).

TensorFlow is a highly flexible and efficient framework for numerical computation, capable of running on multiple platforms (CPU/GPU).

TensorFlow is a highly flexible and efficient framework for numerical computation, capable of running on multiple platforms (CPU/GPU).

TensorFlow is a highly flexible and efficient framework for numerical computation, capable of running on multiple platforms (CPU/GPU).

TensorFlow is a highly flexible and efficient framework for numerical computation, capable of running on multiple platforms (CPU/GPU).