Experiment No 1

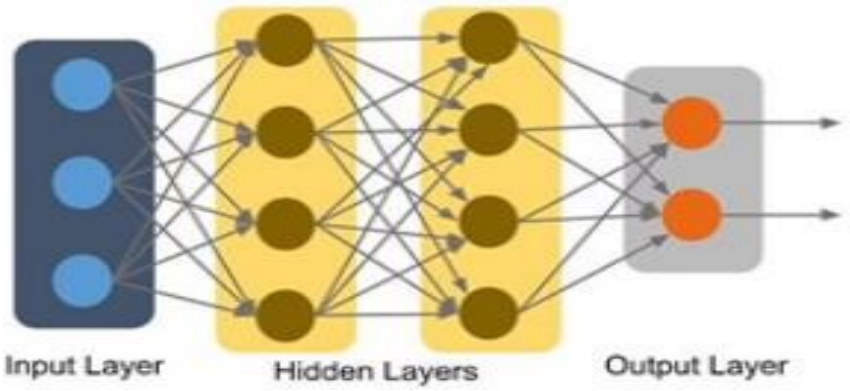
Aim: Study of deep learning packages: Tensorflow,Keras,Theano and PyTorch. Document the distinct featurs and functionality of the packages.

Performance	Understanding	Regularity	Total	Dated Sign of Subject Teacher
03	01	01	05	

Deep Learning

Deep learning is a subset of machine learning, and it works on the structure and functions similarly to the human brain. It learns from data that is unstructured and uses complex algorithms to train a neural net.

We primarily use neural networks in deep learning, which is based on AI. Here, we train networks to recognize text, numbers, images, voice, and so on. Unlike traditional machine learning, the data here is far more complicated, unstructured, and varied, such as images, audio, or text files. One of the core components of deep learning is the neural network, which typically looks like the image shown below:



As seen above, there is an input layer, an output layer, and in between, there are several hidden layers. For any neural network, there would be at least one hidden layer. A deep neural network is one that has more than one hidden layer.

Let us explore the different layers in more detail.

Input Layer

The input layer accepts large volumes of data as input to build the neural network. The data can be in the form of text, image, audio, etc.

Hidden Layer

This layer processes data by performing complex computations and carries out feature extraction. As part of the training, these layers have weights and biases that are continuously

updated until the training process is complete. Each neuron has multiple weights and one bias. After computation, the values are passed to the output layer.

Output Layer

The output layer generates predicted output by applying suitable activation functions. The output can be in the form of numeric or categorical values.

For example, if it is an image classification application, it tells us which class a particular image may belong to. The input can be multiple images, such as cats and dogs. The output can be in the form of binary classification like the number zero for the dog and the number one for the cat.

The network can be extended with multiple neurons on the output side to have many more classes. It can also be used for regression and time series problems.

There are some libraries that are readily available, primarily for performing machine learning and deep learning programming. Some of the most common libraries are as follows:

Tensorflow

TensorFlow is an open-source library developed by Google primarily for deep learning applications. It also supports traditional machine learning. TensorFlow was originally developed for large numerical computations without keeping deep learning in mind. However, it proved to be very useful for deep learning development as well, and therefore Google open-sourced it.

TensorFlow accepts data in the form of multi-dimensional arrays of higher dimensions called tensors. Multi-dimensional arrays are very handy in handling large amounts of data.

TensorFlow works on the basis of data flow graphs that have nodes and edges. As the execution mechanism is in the form of graphs, it is much easier to execute TensorFlow code in a distributed manner across a cluster of computers while using GPUs.

Why TensorFlow?

TensorFlow Offers Both C++ and Python API's

Before the development of libraries, the coding mechanism for machine learning and deep learning was much more complicated. This library provides a high-level API, and complex coding isn't needed to prepare a neural network, configure a neuron, or program a neuron. The library completes all of these tasks. TensorFlow also has integration with Java and R.

TensorFlow Supports Both CPUs and GPUs Computing Devices

Deep learning applications are very complicated, with the training process requiring a lot of computation. It takes a long time because of the large data size, and it involves several iterative processes, mathematical calculations, matrix multiplications, and so on. If you perform these activities on a normal Central Processing Unit (CPU), typically it would take much longer.

Graphical Processing Units (GPUs) are popular in the context of games, where you need the screen and image to be of high resolution. GPUs were originally designed for this purpose. However, they are being used for developing deep learning applications as well.

One of the major advantages of TensorFlow is that it supports GPUs, as well as CPUs. It also has a faster compilation time than other deep learning libraries, like Keras and Torch.

How TensorFlow Works

TensorFlow allows you to create dataflow graphs that describe how data moves through a graph. The graph consists of nodes that represent a mathematical operation. A connection or edge between nodes is a multidimensional data array. It takes inputs as a multi-dimensional array where you can construct a flowchart of operations that can be performed on these inputs.

TensorFlow Architecture

Tensorflow architecture works in three significant steps:

Data pre-processing - structure the data and brings it under one limiting value

Building the model - build the model for the data

Training and estimating the model - use the data to train the model and test it with unknown data

Where Can Tensorflow Run?

TensorFlow requirements can be classified into the development phase (training the model) and run phase (running the model on different platforms). The model can be trained and used on GPUs as well as CPUs. Once the model has been trained, you can run it on:

Desktop (Linux, Windows, macOS)

Mobile devices (iOS and Android)

Cloud as a web service

Keras

Keras is an open source deep learning framework for python. It has been developed by an artificial intelligence researcher at Google named **Francois Chollet**. Leading organizations like Google, Square, Netflix, Huawei and Uber are currently using Keras.

Keras runs on top of open source machine libraries like TensorFlow, Theano or Cognitive Toolkit (CNTK).

Keras is based on minimal structure that provides a clean and easy way to create deep learning models based on TensorFlow or Theano. Keras is designed to quickly define deep learning models. Well, Keras is an optimal choice for deep learning applications.

Features

Keras leverages various optimization techniques to make high level neural network API easier and more performant. It supports the following features −

- Consistent, simple and extensible API.
- Minimal structure - easy to achieve the result without any frills.
- It supports multiple platforms and backends.
- It is user friendly framework which runs on both CPU and GPU.
- Highly scalability of computation.

Benefits

Keras is highly powerful and dynamic framework and comes up with the following advantages −

- Larger community support.
- Easy to test.
- Keras neural networks are written in Python which makes things simpler.
- Keras supports both convolution and recurrent networks.
- Deep learning models are discrete components, so that, you can combine into many ways.

Theano

Theano is a Python library also known as the grandfather of deep learning libraries is popular among researchers. It is a compiler for manipulating and analyzing mathematical expressions, particularly matrix-valued expressions. Computations in Theano are written in a NumPy-like syntax and built to run quickly on either CPU or GPU architectures. Theano python library has garnered attention in the machine learning community as it lets you build wrapper libraries or deep learning or neural network models efficiently at high speeds.

Why Theano for Deep Learning?

- A smart deep learning framework that lets you perform data-intensive computations faster than a CPU or a GPU with efficient native libraries like BLAS.
- Evaluating expressions is much faster because of the dynamic C code it generates.
- Creates Symbolic graphs for computing gradients automatically.
- Provides stable means to optimize and evaluate unstable expressions.

PyTorch

PyTorch is an open source machine learning library for Python and is completely based on Torch. It is primarily used for applications such as natural language processing. PyTorch is developed by Facebook's artificial-intelligence research group along with Uber's "Pyro" software for the concept of in-built probabilistic programming.

Originally, PyTorch was developed by Hugh Perkins as a Python wrapper for the LusJIT based on Torch framework. There are two PyTorch variants.

PyTorch redesigns and implements Torch in Python while sharing the same core C libraries for the backend code. PyTorch developers tuned this back-end code to run Python efficiently. They also kept the GPU based hardware acceleration as well as the extensibility features that made Lua-based Torch.