

**Experiment No 1****Installation of Anaconda and introduction to Tensorflow and Keras****Objective:**

At the end of this practical session, student will be able to set up an environment for developing Deep Learning models.

**Theory:****A. Anaconda**

Anaconda is a free and open-source distribution of the Python for scientific computing (data science, machine learning applications, large-scale data processing), that aims to simplify package management and deployment. In it, Package versions are managed by the package management system conda. The Anaconda distribution includes more than 1500 popular data-science packages.

Anaconda distribution comes with Anaconda Navigator, which is alternative to the command line interface (CLI). Anaconda Navigator is a desktop graphical user interface (GUI) that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

**B. Tensorflow**

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, and community resources that lets developers to build and deploy ML powered applications easily. It has an open source software library used for numerical computations using data flow graphs. Nodes in graph represents computations or operations while edges represents data or tensors.

Tensorflow takes two steps for building a graph based model.

1. Specify the operation and data to develop a static graph.
2. Execute the graph to get the final result.

Execution can be done on any intermediate node. It does not execute portion of graph that is not in the path of node i.e. TensorFlow calculates only that portion of the graph, which is required.

**C. Keras**

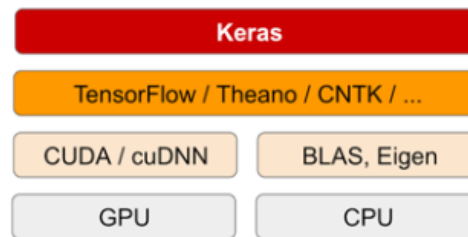
Keras is a deep-learning framework for Python that provides a convenient way to define and train almost any kind of deep-learning model

**Key features:**

- It allows the same code to run seamlessly on CPU or GPU.
- It has a user-friendly API that makes it easy to quickly prototype deep-learning models.
- It has built-in support for convolutional networks (for computer vision), recurrent networks (for sequence processing), and any combination of both.

- It supports arbitrary network architectures: multi-input or multi-output models, layer sharing, model sharing, and so on.

Keras provides high-level building blocks for developing deep-learning models but it doesn't handle low-level operations such as tensor manipulation and differentiation. So Keras relies on a specialized, well-optimized tensor library to do tensor manipulation, serving as the backend engine of Keras. Currently it supports three backends: TensorFlow, the Theano, and the Microsoft Cognitive Toolkit (CNTK). Following figure shows relationship between hardware and different libraries.



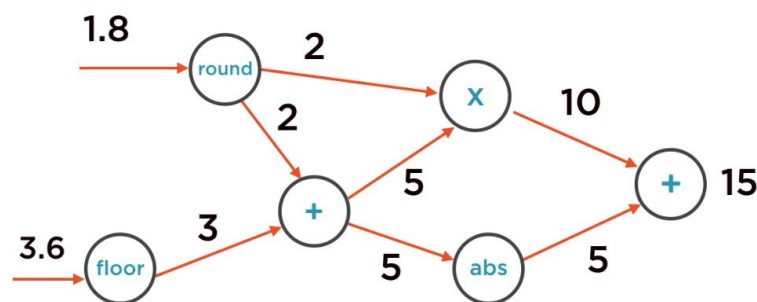
To use Keras with tensorflow as backend, we have to install Tensorflow package first and then Keras packages separately.

#### Keyword:

Anaconda, Tensorflow, Keras

#### Procedure:

1. Install Anaconda on your machine.
2. Create new environment in anaconda using anaconda navigator.
3. Install a tensorflow library in that environment.
4. Develop following graph using tensorflow.



5. Execute graph using `tf.Session()`.
6. Install Keras library in same environment.
7. Import keras and execute cell in jupyter notebook. Check the message 'using Tensorflow backend' after execution.