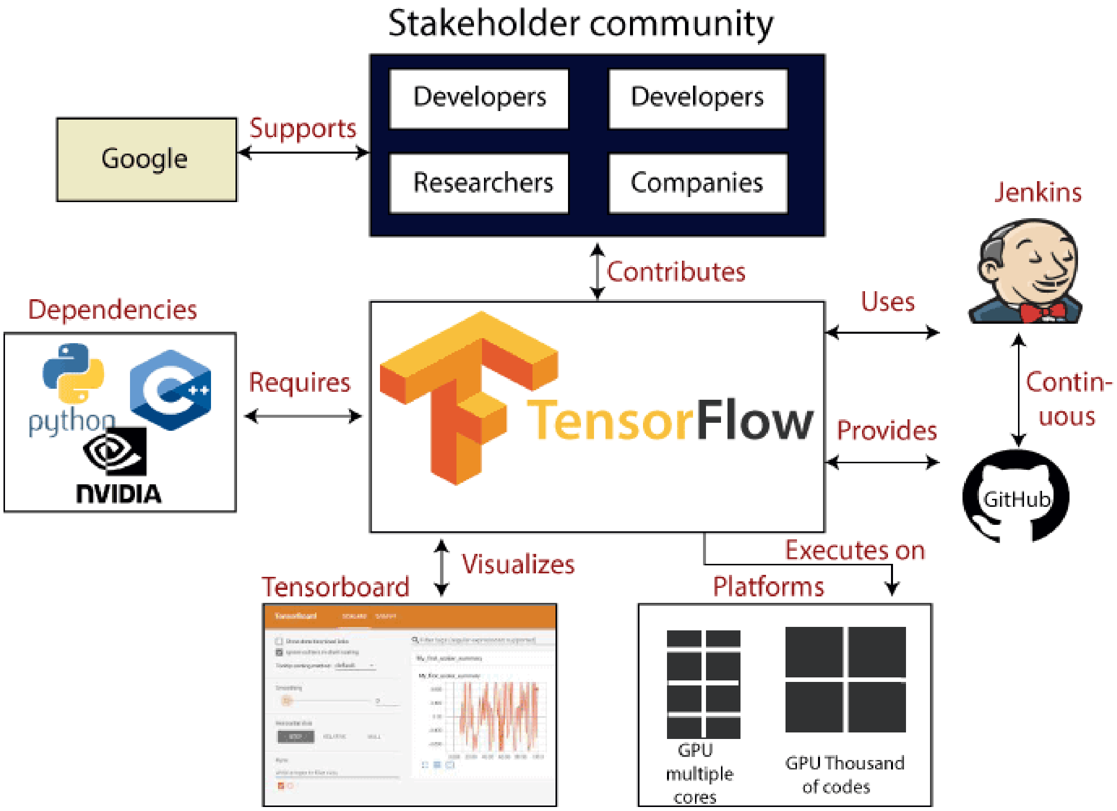
Experiment:1. TensorFlow Basics and Examples Date: 12/12/2023

TensorFlow basics:

TensorFlow is an end-to-end open-source platform for machine learning library developed by the Google Bring team. It has an ecosystem of tools, libraries, and community resources. It is widely used for building and training machine learning and deep learning models.

Tensor is the fundamental building block in TensorFlow. A tensor is a multi-dimensional array with a uniform data type. It can be a scalar (0-D tensor), a vector (1-D tensor), a matrix (2-D tensor), or have more dimensions.



Scalars: Scalars are just a single number. For example, temperature, which is denoted by just one number.

Vectors: Vectors are an array of numbers. The numbers are arranged in order and we can identify each individual number by its index in that ordering. We can think of vectors as identifying points in space, with each element giving the coordinate along a different axis. In simple terms, a vector is an arrow representing a quantity that has both magnitude and direction wherein the length of the arrow represents the magnitude and the orientation tells you the direction. For example, wind, which has a direction and magnitude.

Matrices: A matrix is a 2D-array of numbers, so each element is identified by two indices instead of just one. If a real valued matrix *A* has a height of *m* and a width of *n*, then we say that A in R^ (m x n). We identify the elements of the matrix as A_(m,n) where *m* represents the row and *n* represents the column.

Tensor: A tensor is simply an n-dimensional array of numbers. For example, a vector is a onedimensional tensor, a matrix is a two-dimensional tensor, and an image is a three-dimensional tensor (width, height, and depth).

Example: Numerical Methods to solve matrix problems in Python

Program:

```
import os
os.environ['TF_CPP_MIN_LOG_LEVEL']='2' import
tensorflow as tfx=tf.constant(89.0)
x=tf.constant(x,shape=(1,2),dtype=tf.float32)print(x.shape)
```

Output:

```
(1, 2)
```

```
y=tf.constant([[1,2,5],[55,7,99]])
print(y.shape)
```

Output:

```
(2, 3)
```

```
x=tf.zeros((2,3))x
```

Output:

```
<tf.Tensor: shape=(2, 3), dtype=float32, numpy=
array([[0., 0., 0.],
[0., 0., 0.]], dtype=float32)>
```

```
x=tf.eye(6)
x
```

Output:

```
<tf.Tensor: shape=(6, 6), dtype=float32, numpy=
                    array([[1., 0., 0., 0., 0.,
                0.],             [0., 1., 0., 0.,
                                       0., 0.],
[0., 0., 1., 0., 0., 0.],
[0., 0., 0., 1., 0., 0.],
[0., 0., 0., 0., 1., 0.],
[0., 0., 0., 0., 0., 1.]], dtype=float32)>
```

```
x=tf.random.normal((3,3),mean=8,stddev=0.23)
x
```

Output:

```
<tf.Tensor: shape=(3, 3), dtype=float32, numpy=
array([[7.3518577, 8.063381 , 8.308786 ],
[7.773917 , 8.209878 , 7.916413 ],

[7.9034214, 7.9605427, 8.335818 ]], dtype=float32)>
```

```
import os os.environ['TF_CPP_MIN_LOG_LEVEL']='2'import
tensorflow as tf
##indexingx=tf.constant([1,4,6,633,4,2])print(x[:]) #print all
elements print(x[1:]) #print from 1st index
print(x[1:4:2]) # print from 1 to 4 index skiping evering 2 elementprint(x[1::-1]) # print reverse
from 1st indexprint(x[::-4])
#print in                reverse upto 4th index
```

Output:

```
tf.Tensor([ 1                4        6  633        4        2],
tf.Tensor([   4                                shape=(6,),  dtype=int32)
                      6  6334     2],  shape=(5,),  dtype=int32)
tf.Tensor([ 4 633], shape=(2,), dtype=int32)  tf.Tensor([4 1],
shape=(2,),        dtype=int32)                      4],   shape=(2,),
                   tf.Tensor([2dtype=int32)
```