Introduction to Tensorflow

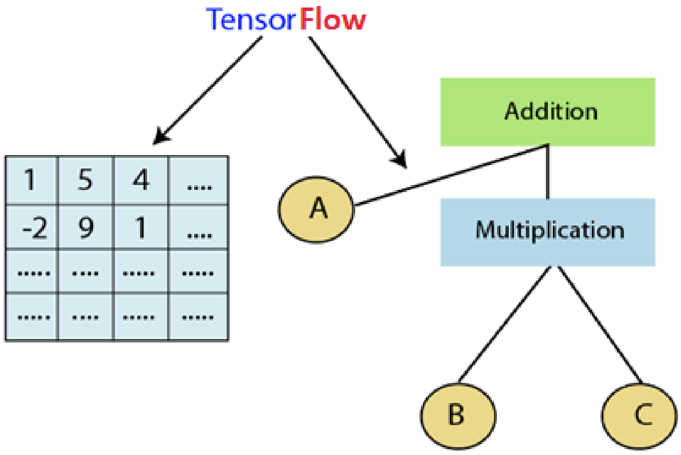
TensorFlow is a popular framework of **machine learning** and **deep learning**. It is **a free** and **open-source** library which is released on **9 November 2015** and developed by **Google Brain Team**. It is entirely based on Python programming language and use for numerical computation and data flow, which makes machine learning faster and easier.

TensorFlow can train and run the deep neural networks for image recognition, handwritten digit classification, recurrent neural network, **word embedding**, **natural language processing**, video detection, and many more. TensorFlow is run on multiple **CPU**s or **GPU**s and also mobile operating systems.

The word TensorFlow is made by two words, i.e., Tensor and Flow

1. **Tensor** is a multidimensional array
2. **Flow** is used to define the flow of data in operation.

TensorFlow is used to define the flow of data in operation on a multidimensional array or Tensor.



1	5	4	....
-2	9	1	....
.....	....	.....	.....
.....	....	.....	.....

History of TensorFlow

Many years ago, deep learning started to exceed all other machine learning algorithms when giving extensive data. Google has seen it could use these deep neural networks to upgrade its services:

- Google search engine
- Gmail
- Photo

They build a framework called TensorFlow to permit researchers and developers to work together in an **AI** model. Once it approved and scaled, it allows lots of people to use it.

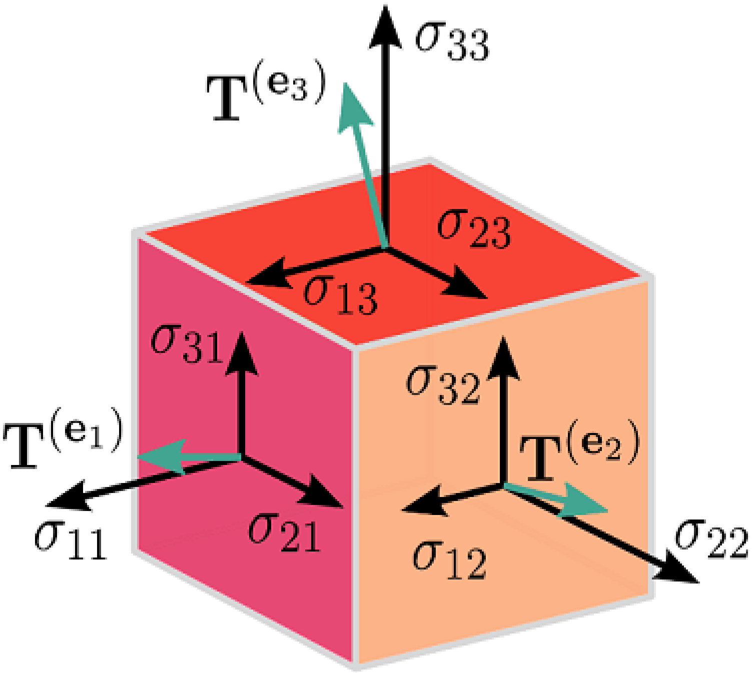
It was first released in 2015, while the first stable version was coming in **2017**. It is an open- source platform under Apache Open Source License. We can use it, modify it, and reorganize the revised version for free without paying anything to Google.

Components of TensorFlow

Tensor

The name TensorFlow is derived from its core framework, "**Tensor**." A tensor is a vector or a matrix of n-dimensional that represents all type of data. All values in a tensor hold similar data type with a known shape. The shape of the data is the dimension of the matrix or an array.

A tensor can be generated from the input data or the result of a computation. In TensorFlow, all operations are conducted inside a graph. The group is a set of calculation that takes place successively. Each transaction is called an op node are connected.



Graphs

TensorFlow makes use of a graph framework. The chart gathers and describes all the computations done during the training.

Advantages

- It was fixed to run on multiple CPUs or GPUs and mobile operating systems.
- The portability of the graph allows to conserve the computations for current or later use. The graph can be saved because it can be executed in the future.

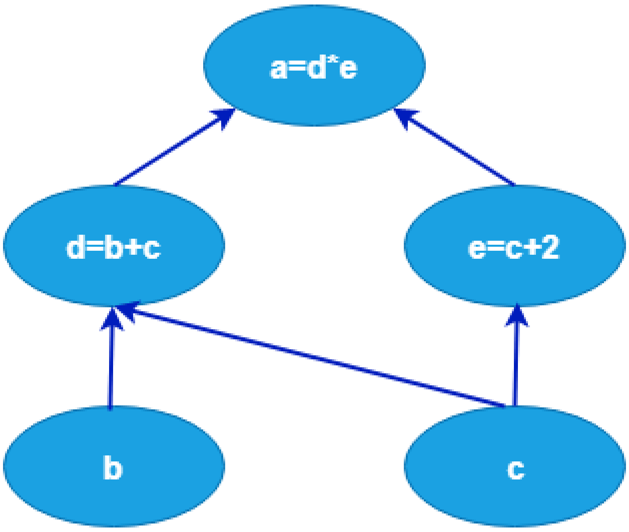
- All the computation in the graph is done by connecting tensors together.

Consider the following expression a= (b+c)*(c+2)

We can break the functions into components given below:

d=b+c
e=c+2
a=d*e

Now, we can represent these operations graphically below:



Session

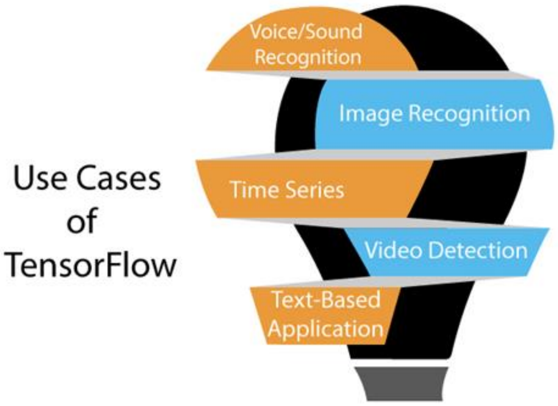
A session can execute the operation from the graph. To feed the graph with the value of a tensor, we need to open a session. Inside a session, we must run an operator to create an output.

Why is TensorFlow popular?

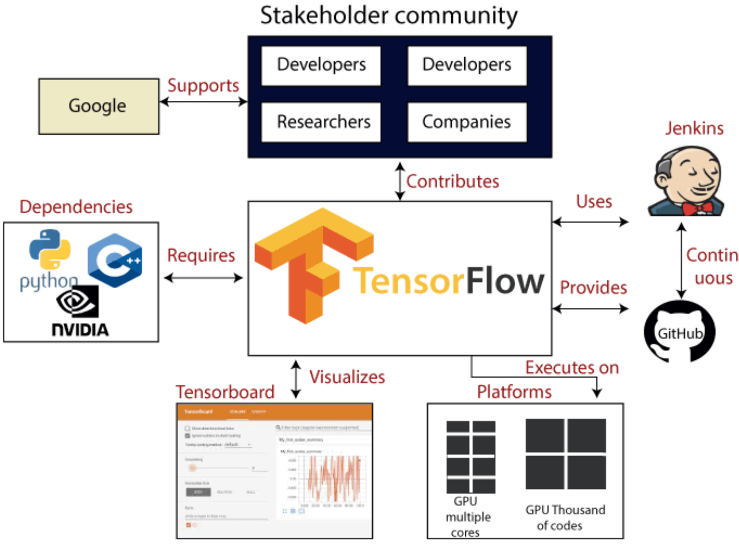
TensorFlow is the better library for all because it is accessible to everyone. TensorFlow library integrates different **API** to create a scale deep learning architecture like **CNN (Convolutional Neural Network)** or **RNN (Recurrent Neural Network)**.

TensorFlow is based on graph computation; it can allow the developer to create the construction of the neural network with Tensorboard. This tool helps debug our program. It runs on CPU (Central Processing Unit) and GPU (Graphical Processing Unit).

Use Cases/Applications of TensorFlow



TensorFlow provides amazing functionalities and services when compared to other popular deep learning frameworks. TensorFlow is used to create a large-scale **neural network** with many layers.

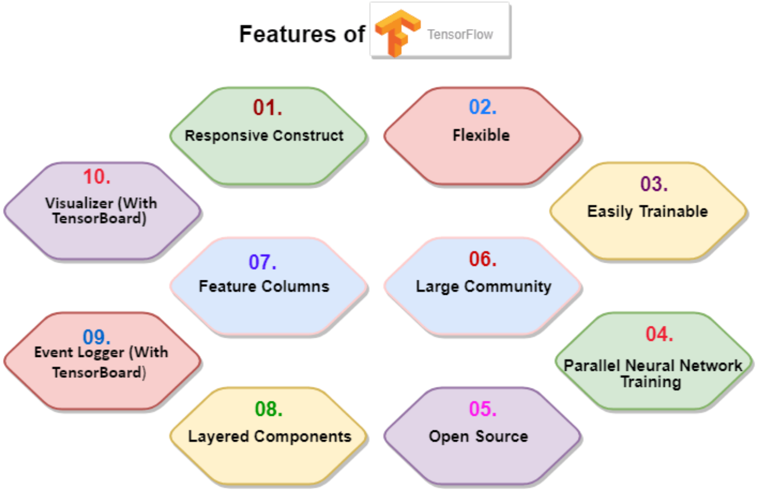


It is mainly used for deep learning or machine learning problems such as **Classification**, **Perception**, **Understanding**, **Discovering Prediction**, and **Creation**.

Features of TensorFlow

TensorFlow has an interactive **multiplatform** programming interface which is scalable and reliable compared to other deep learning libraries which are available.

These features of TensorFlow will tell us about the popularity of TensorFlow.



1. Responsive Construct

We can visualize each part of the graph, which is not an option while using **Numpy** or **SciKit**. To develop a deep learning application, firstly, there are two or three components that are required to create a deep learning application and need a programming language.

2. Flexible

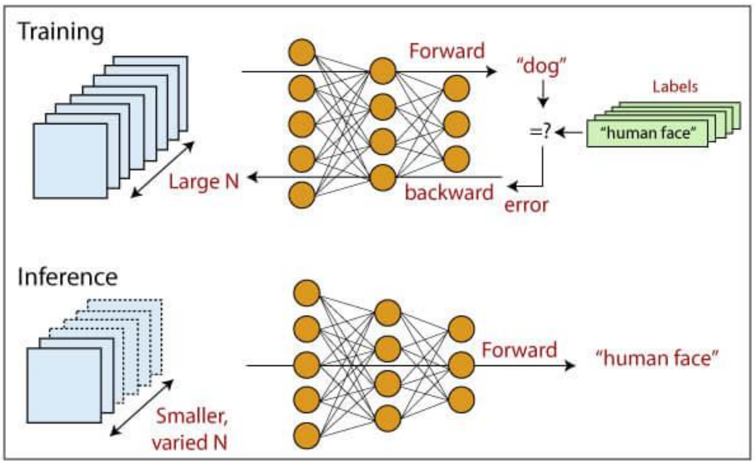
It is one of the essential TensorFlow Features according to its operability. It has modularity and parts of it which we want to make standalone.

3. Easily Trainable

It is easily trainable on CPU and for GPU in distributed computing.

4. Parallel Neural Network Training

TensorFlow offers to the pipeline in the sense that we can train multiple neural networks and various **GPUs**, which makes the models very efficient on large-scale systems.



5. Large Community

Google has developed it, and there already is a large team of software engineers who work on stability improvements continuously.

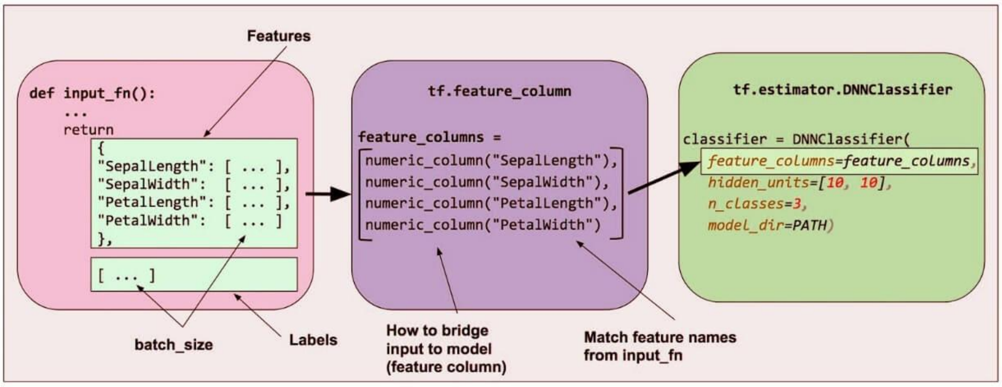
6. Open Source

The best thing about the machine learning library is that it is open source so anyone can use it as much as they have internet connectivity. So, people can manipulate the library and come up with a fantastic variety of useful products. And it has become another **DIY** community which has a massive forum for people getting started with it and those who find it hard to use it.

7. Feature Columns

TensorFlow has feature columns which could be thought of as intermediates between raw data and estimators; accordingly, **bridging** input data with our model.

The feature below describes how the feature column is implemented.



Features
def input_fn():
...
return
{
"SepalLength": [...],
"SepalWidth": [...],
"PetalLength": [...],
"PetalWidth": [...]
},
[...]
tf.feature_column
feature_columns =
numeric_column("SepalLength"),
numeric_column("SepalWidth"),
numeric_column("PetalLength"),
numeric_column("PetalWidth")
tf.estimator.DNNClassifier
classifier = DNNClassifier(
feature_columns=feature_columns,
hidden_units=[10, 10],
n_classes=3,
model_dir=PATH)
batch_size
Labels
How to bridge input to model (feature column)
Match feature names from input_fn

8. Availability of Statistical Distributions

This library provides distributions functions including Bernoulli, Beta, Chi2, Uniform, Gamma, which are essential, especially where considering probabilistic approaches such as Bayesian models.

9. Layered Components

TensorFlow produces layered operations of weight and biases from the function such as tf.contrib.layers and also provides batch normalization, convolution layer, and dropout layer. So **tf.contrib.layers.optimizers** have optimizers such as **Adagrad**, **SGD**, **Momentum** which are often used to solve optimization problems for numerical analysis.

10. Visualizer (With TensorBoard)

We can inspect a different representation of a model and make the changed necessary while debugging it with the help of TensorBoard.

11.Event Logger (With TensorBoard)

It is just like UNIX, where we use **tail - f** to monitor the output of tasks at the cmd. It checks, logging events and summaries from the graph and production with the TensorBoard.