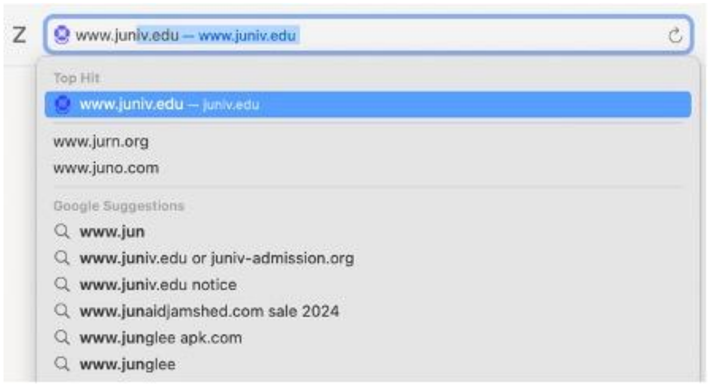
TensorFlow

What is TensorFlow?

TensorFlow is a library developed by the Google Brain Team to accelerate machine learning and deep neural network research. The most famous deep learning library in the world is Google's TensorFlow. Google product uses machine learning in all of its products to improve the search engine, translation, image captioning or recommendations. It was built to run on multiple CPUs or GPUs and even mobile operating systems, and it has several wrappers in several languages like Python, C++ or Java.



www.juniv.edu — www.juniv.edu

Top Hit

www.juniv.edu — juniv.edu

www.jurn.org

www.juno.com

Google Suggestions

www.jun

www.juniv.edu or juniv-admission.org

www.juniv.edu notice

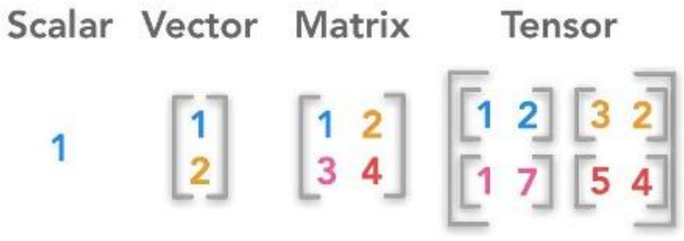
www.junaidjamshed.com sale 2024

www.junglee apk.com

www.junglee

For example, Google users can experience a faster and more refined the search with AI. If the user types a keyword a the search bar, Google provides a recommendation about what could be the next word.

It is called TensorFlow because it takes input as a multi-dimensional array, also known as tensors. Users can construct a sort of flowchart of operations i.e. Graph that users want to perform on that input. The input goes in at one end, and then it flows through this system of multiple operations and comes out the other end as output. This is why it is called TensorFlow because the tensor goes in it flows through a list of operations, and then it comes out the other side.



History of TensorFlow

A couple of years ago, deep learning started to outperform all other machine learning algorithms when giving a massive amount of data. Google saw it could use these deep neural networks to improve its services:

- Gmail
- Photo
- Google search engine

They build a framework called TensorFlow to let researchers and developers work together on an AI model. Once developed and scaled, it allows lots of people to use it. It was first made public in late 2015, while the first stable version appeared in 2017. It is open source under Apache Open Source license. Anybody can use it, modify it and redistribute the modified version for a fee without paying anything to Google.

What are the Phases of TensorFlow?

TensorFlow Phases:

Development Phase: This is when you train the mode. Training is usually done on Desktop or laptop.

Run Phase or Inference Phase: Once training is done TensorFlow can be run on many different platforms. Users can run it on

- Desktop running Windows, macOS or Linux
- Cloud as a web service,
- Mobile devices like iOS and Android.

Users can train it on multiple machines then he can run it on a different machine once he has the trained model.

The model can be trained and used on GPUs as well as CPUs. Deep learning relies on a lot of matrix multiplication. TensorFlow is very fast at computing the matrix multiplication because it is written in C++. Although it is implemented in C++, TensorFlow can be accessed and controlled by other languages mainly, Python. Finally, a significant feature of TensorFlow is the TensorBoard. The TensorBoard enables to monitor graphically and visually what TensorFlow is doing.

N.B.: GPUs were initially designed for video games. In late 2010, Stanford researchers found that GPU was also very good at matrix operations and algebra so that it makes them very fast for doing these kinds of calculations.

What are the components of TensorFlow?

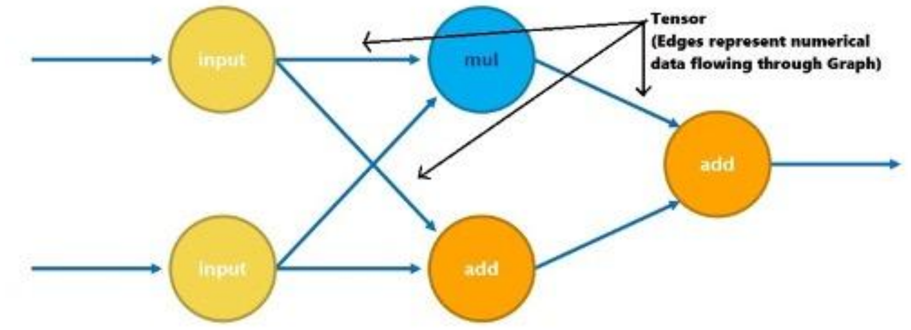
Components of TensorFlow

Tensor: Tensorflow's name is directly derived from its core framework: Tensor. In Tensorflow, all the computations involve tensors. A tensor is a vector or matrix of n-dimensions that represents all types of data. All values in a tensor hold identical data type with a known shape. The shape of the data is the dimensionality of the matrix or array.

A tensor can be originated from the input data or the result of a computation. In TensorFlow, all the operations are conducted inside a graph. The graph is a set of computation that takes place successively. Each operation is called an op node and are connected to each other. The graph outlines the ops and connections between the nodes. However, it does not display the values. The edge of the nodes is the tensor, i.e., a way to populate the operation with data.

Graphs: TensorFlow makes use of a graph framework. The graph gathers and describes all the series computations done during the training. The graph has lots of advantages:

- It was done to run on multiple CPUs or GPUs and even mobile operating system
- The portability of the graph allows to preserve the computations for immediate or later use. The graph can be saved to be executed in the future.
- All the computations in the graph are done by connecting tensors together A tensor has a node and an edge. The node carries the mathematical operation and produces an endpoints outputs. The edges explain the input/output relationships between nodes.



Why is TensorFlow popular?

Reasons behind the popularity of TensorFlow

TensorFlow is the best library of all because it is built to be accessible for everyone. It is now-a-days is a popular due to its following advantages:

- Tensorflow library incorporates different API to built at scale deep learning architecture like CNN or RNN.
- TensorFlow is based on graph computation; it allows the developer to visualize the construction of the neural network with Tensorboad. This tool is helpful to debug the program.
- Finally, Tensorflow is built to be deployed at scale. It runs on CPU and GPU.
- Tensorflow attracts the largest popularity on GitHub compare to the other deep learning framework.

Prominent Algorithms supported by TensorFlow

The latest stable version of TensorFlow is 2.19.0, released on March 12, 2025. Algorithm tha are included in TensorFlow are:

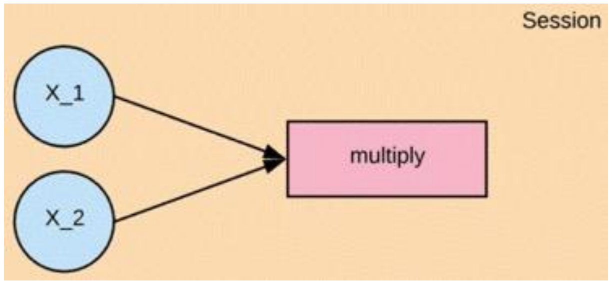
- Supervised Learning Algorithms: Linear Regression, Logistic Regression, Support Vector Machines (SVM), Decision Trees, Random Forests, k-Nearest Neighbors (k-NN), Gradient Boosting (e.g., XGBoost via integration)
- Deep Learning Algorithms: Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Transformer Models, Autoencoders, Generative Adversarial Networks (GANs)
- Unsupervised Learning Algorithms: k-Means Clustering, Hierarchical Clustering, Principal Component Analysis (PCA), t-SNE (for visualization), Self-Organizing Maps (SOMs), DBSCAN
- Reinforcement Learning (via TensorFlow Agents): Q-Learning, Deep Q Networks (DQN), Policy Gradient Methods, Proximal Policy Optimization (PPO), Actor-Critic Methods (A2C, A3C)

Simple TensorFlow Example

import numpy as np

import tensorflow as tf

Let 's create a computational graph that multiplies two numbers together.Tensorflow will create a node to connect the operation. In our example, it is called multiply. When the graph is determined, Tensorflow computational engines will multiply together X_1 and X_2.



Finally, we will run a TensorFlow session that will run the computational graph with the values of X_1 and X_2 and print the result of the multiplication. When we create a node in Tensorflow, we have to choose what kind of node to create. The X1 and X2 nodes will be a placeholder node. The placeholder assigns a new value each time we make a calculation. We will create them as a TF dot placeholder node.

Step 1: Define the variable

X_1 = tf.placeholder(tf.float32, name = "X_1")

X_2 = tf.placeholder(tf.float32, name = "X_2")

Step 2: Define the computation

multiply = tf.multiply(X_1, X_2, name = "multiply")

Now we can define the node that does the multiplication operation. In Tensorflow we can do that by creating a tf.multiply node.

Step 3: Execute the operation

To execute operations in the graph, we have to create a session. In Tensorflow, it is done by tf.Session(). After creating or having a session we can ask the session to run operations on our computational graph by calling session. To run thecomputation, we need to use run.

When the addition operation runs, it is going to see that it needs to grab the values of the X_1 and X_2 nodes, so we also need to feed in values for X_1 and X_2. We can do that by supplying a parameter called feed_dict. We pass the value 1,2,3 for X_1 and 4,5,6 for X_2.

```
X_1 = tf.placeholder(tf.float32, name = "X_1")

X_2 = tf.placeholder(tf.float32, name = "X_2")

multiply = tf.multiply(X_1, X_2, name = "multiply")

with tf.Session() as session:

    result = session.run(multiply, feed_dict={X_1:[1,2,3], X_2: [4,5,6]})

    print(result)

[ 4. 10. 18.]
```