Unit-5

A neural network is **a method in artificial intelligence that teaches computers to process data in a way that is inspired by the human brain**. It is a type of machine learning process, called deep learning that uses interconnected nodes or neurons in a layered structure that resembles the human brain. It creates an adaptive system that computers use to learn from their mistakes and improve continuously. Thus, artificial neural networks attempt to solve complicated problems, like summarizing documents or recognizing faces, with greater accuracy.

Why are neural networks important?

Neural networks can help computers make intelligent decisions with limited human assistance. This is because they can learn and model the relationships between input and output data that are nonlinear and complex. For instance, they can do the following tasks.

What are neural networks used for?

Neural networks have several use cases across many industries, such as the following:

- Medical diagnosis by medical image classification
- Targeted marketing by social network filtering and behavioral data analysis
- Financial predictions by processing historical data of financial instruments
- Electrical load and energy demand forecasting
- Process and quality control
- Chemical compound identification

How do neural networks work?

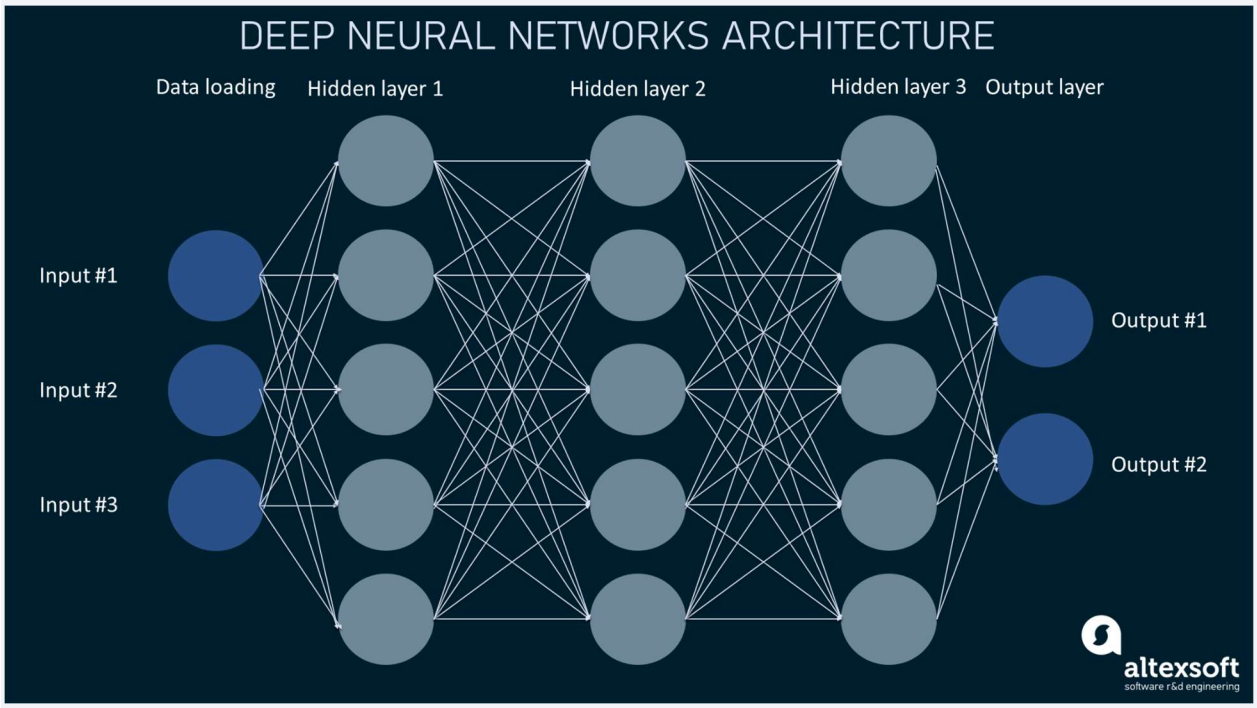
The human brain is the inspiration behind neural network architecture. Human brain cells, called neurons, form a complex, highly interconnected network and send electrical signals to each other to help humans process information. Similarly, an artificial neural network is made of artificial neurons that work together to solve a problem. Artificial neurons are software modules, called nodes, and artificial neural networks are software programs or algorithms that, at their core, use computing systems to solve mathematical calculations.

Simple neural network architecture

A basic neural network has interconnected artificial neurons in three layers:

Input Layer

Information from the outside world enters the artificial neural network from the input layer. Input nodes process the data, analyze or categorize it, and pass it on to the next layer.



Hidden Layer

Hidden layers take their input from the input layer or other hidden layers. Artificial neural networks can have a large number of hidden layers. Each hidden layer analyzes the output from the previous layer, processes it further, and passes it on to the next layer.

Output Layer

The output layer gives the final result of all the data processing by the artificial neural network. It can have single or multiple nodes. For instance, if we have a binary (yes/no) classification problem, the output layer will have one output node, which will give the result as 1 or 0. However, if we have a multi-class classification problem, the output layer might consist of more than one output node.

Deep neural network architecture

Deep neural networks, or deep learning networks, have several hidden layers with millions of artificial neurons linked together. A number, called weight, represents the connections between one node and another. The weight is a positive number if one node excites another, or negative if one node suppresses the other. Nodes with higher weight values have more influence on the other nodes.
Theoretically, deep neural networks can map any input type to any output type. However, they also need much more training as compared to other machine learning methods. They need millions of examples of training data rather than perhaps the hundreds or thousands that a simpler network might need.

What are the types of neural networks?

Artificial neural networks can be categorized by how the data flows from the input node to the output node. Below are some examples:

Feedforward neural networks

Feedforward neural networks process data in one direction, from the input node to the output node. Every node in one layer is connected to every node in the next layer. A feedforward network uses a feedback process to improve predictions over time.

Backpropagation algorithm

Artificial neural networks learn continuously by using corrective feedback loops to improve their predictive analytics. In simple terms, you can think of the data flowing from the input node to the output node through many different paths in the neural network. Only one path is the correct one that maps the input node to the correct output node. To find this path, the neural network uses a feedback loop, which works as follows:

1. Each node makes a guess about the next node in the path.
2. It checks if the guess was correct. Nodes assign higher weight values to paths that lead to more correct guesses and lower weight values to node paths that lead to incorrect guesses.
3. For the next data point, the nodes make a new prediction using the higher weight paths and then repeat Step 1.

Convolutional neural networks

The hidden layers in convolutional neural networks perform specific mathematical functions, like summarizing or filtering, called convolutions. They are very useful for image classification because they can extract relevant features from images that are useful for image recognition and classification. The new form is easier to process without losing features that are critical for making a good prediction. Each hidden layer extracts and processes different image features, like edges, color, and depth.

How to train neural networks?

Neural network training is the process of teaching a neural network to perform a task. Neural networks learn by initially processing several large sets of labeled or unlabeled data. By using these examples, they can then process unknown inputs more accurately.

Supervised learning

In supervised learning, data scientists give artificial neural networks labeled datasets that provide the right answer in advance. For example, a deep learning network training in facial recognition initially processes hundreds of thousands of images of human faces, with various terms related to ethnic origin, country, or emotion describing each image.

The neural network slowly builds knowledge from these datasets, which provide the right answer in advance. After the network has been trained, it starts making guesses about the ethnic origin or emotion of a new image of a human face that it has never processed before.

What is deep learning in the context of neural networks?

Artificial intelligence is the field of computer science that researches methods of giving machines the ability to perform tasks that require human intelligence. Machine learning is an artificial intelligence technique that gives computers access to very large datasets and teaches them to learn from this data. Machine learning software finds patterns in existing data and applies those patterns to new data to make intelligent decisions. Deep learning is a subset of machine learning that uses deep learning networks to process data.

Machine learning vs. deep learning

Traditional machine learning methods require human input for the machine learning software to work sufficiently well. A data scientist manually determines the set of relevant features that the software must analyze. This limits the software's ability, which makes it tedious to create and manage.

On the other hand, in deep learning, the data scientist gives only raw data to the software. The deep learning network derives the features by itself and learns more independently. It can analyze unstructured datasets like text documents, identify which data attributes to prioritize, and solve more complex problems.

For example, if you were training a machine learning software to identify an image of a pet correctly, you would need to take these steps:

- Find and label thousands of pet images, like cats, dogs, horses, hamsters, parrots, and so on, manually.
- Tell the machine learning software what features to look for so it can identify the image using elimination. For instance, it might count the number of legs, then check for eye shape, ear shape, tail, fur, and so on.
- Manually assess and change the labeled datasets to improve the software's accuracy. For example, if your training set has too many pictures of black cats, the software will correctly identify a black cat but not a white one.
- In deep learning, however, the neural networks would process all the images and automatically determine that they need to analyze the number of legs and the face shape first, then look at the tails last to correctly identify the animal in the image.

What is TensorFlow?

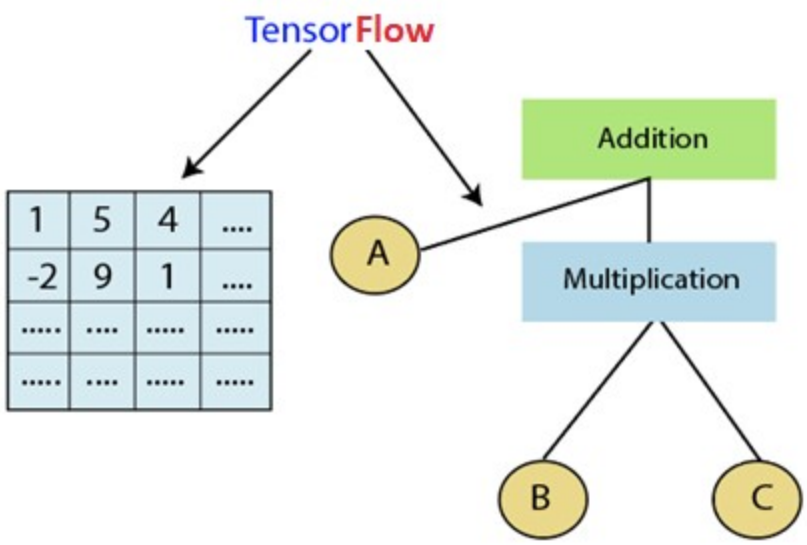
TensorFlow is a popular framework of **machine learning** and **deep learning**. It is **a free** and **open-source** library which is released on **9 November 2015** and developed by **Google Brain Team**. It is entirely based on Python programming language and use for numerical computation and data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



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.

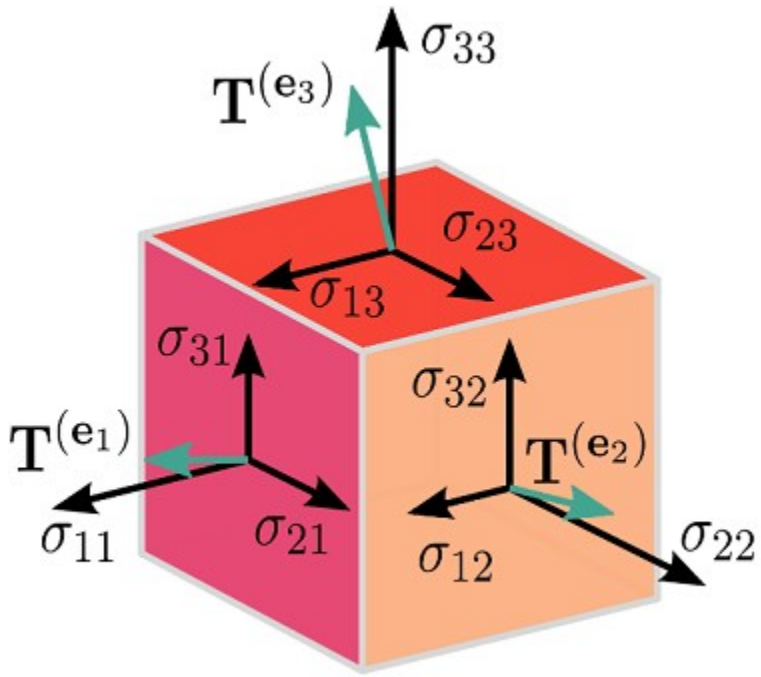
It was first released in 2015, while the first stable version was coming in **2017**. It is an open- source platform under Apache Open Source License. We can use it, modify it, and reorganize the revised version for free without paying anything to Google.

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.

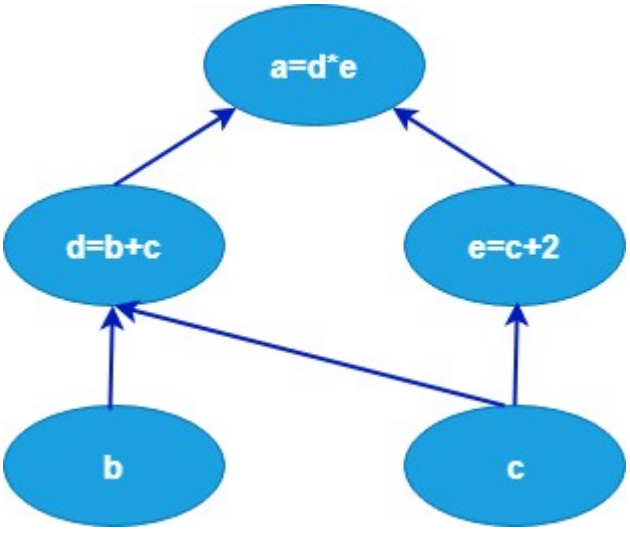
Consider the following expression a= (b+c)*(c+2)

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.

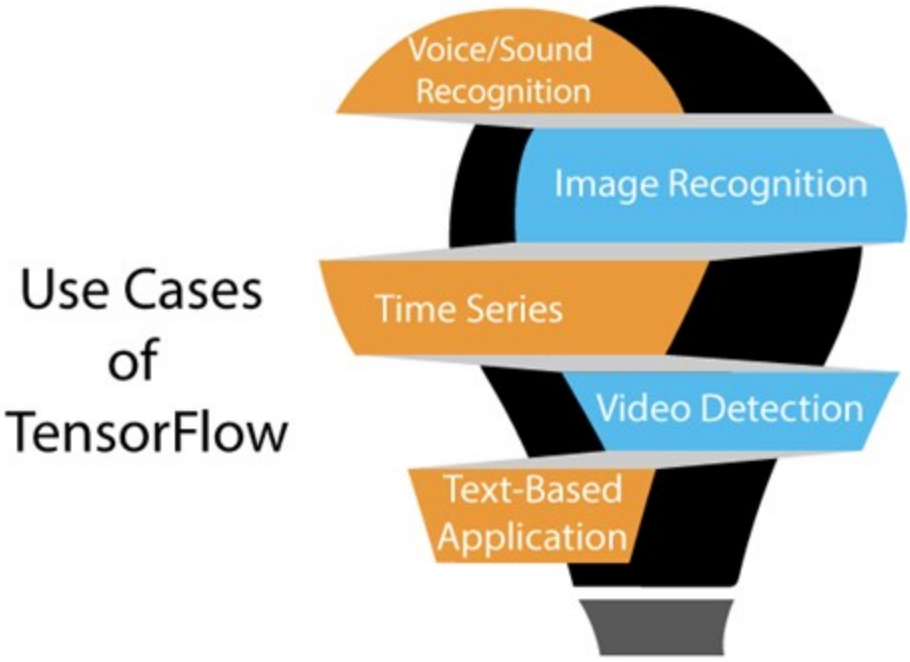
Why is TensorFlow popular?

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)**.

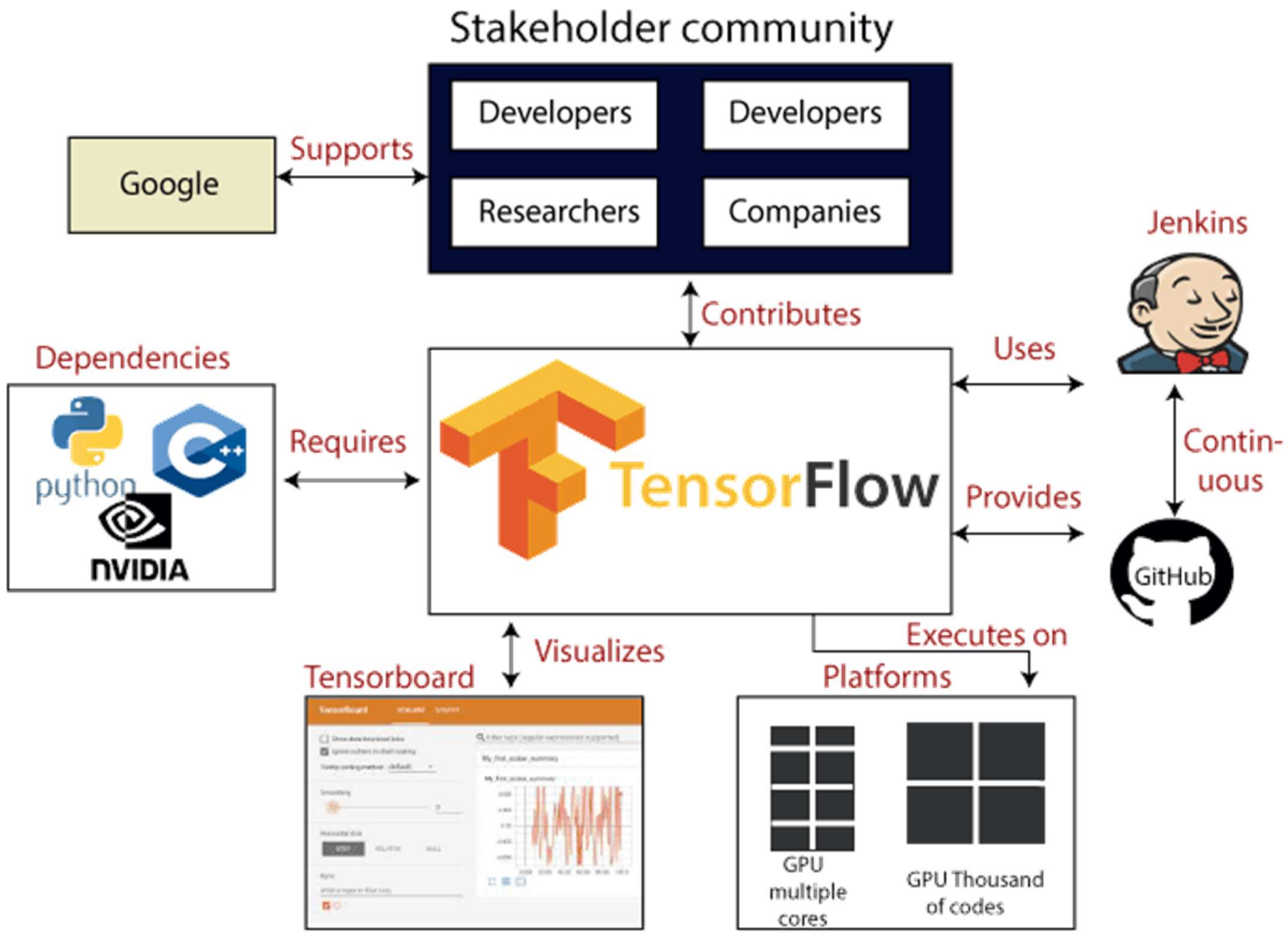
TensorFlow is based on graph computation; it can allow the developer to create the construction of the neural network with Tensorboard. This tool helps debug our program. It runs on CPU (Central Processing Unit) and GPU (Graphical Processing Unit).

TensorFlow *attracts the most considerable popularity on* ***GitHub*** *compare to the other deep learning framework.*

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**.

1. Voice/Sound Recognition

Voice and sound recognition applications are the most-known use cases of deep-learning. If the neural networks have proper input data feed, neural networks are capable of understanding audio signals.

For example:

Voice recognition is used in the Internet of Things, automotive, security, and UX/UI.

Sentiment Analysis is mostly used in customer relationship management **(CRM)**.

Flaw Detection (engine noise) is mostly used in automotive and Aviation.

Voice search is mostly used in customer relationship management (CRM)

2. Image Recognition

Image recognition is the first application that made deep learning and machine learning popular. Telecom, Social Media, and handset manufacturers mostly use image recognition. It is also used for face recognition, image search, motion detection, machine vision, and photo clustering.

For example, image recognition is used to recognize and identify people and objects in from of images. Image recognition is used to understand the context and content of any image.

For object recognition, TensorFlow helps to classify and identify arbitrary objects within larger images.

This is also used in engineering application to identify shape for modeling purpose (**3d** reconstruction from **2d** image) and by Facebook for photo tagging.

For example, deep learning uses TensorFlow for analyzing thousands of photos of cats. So a deep learning algorithm can learn to identify a cat because this algorithm is used to find general features of objects, animals, or people.

3. Time Series

Deep learning is using Time Series algorithms for examining the time series data to extract meaningful statistics. For example, it has used the time series to predict the stock market.

A recommendation is the most common use case for Time Series. **Amazon**, **Google**, **Facebook**, and **Netflix** are using deep learning for the suggestion. So, the deep learning algorithm is used to analyze customer activity and compare it to millions of other users to determine what the customer may like to purchase or watch.

For example, it can be used to recommend us TV shows or movies that people like based on TV shows or movies we already watched.

4. Video Detection

The deep learning algorithm is used for video detection. It is used for motion detection, real-time threat detection in gaming, security, airports, and UI/UX field.

For example, NASA is developing a deep learning network for object clustering of asteroids and orbit classification. So, it can classify and predict NEOs (**Near Earth Objects**).

5. Text-Based Applications

Text-based application is also a popular deep learning algorithm. Sentimental analysis, social media, threat detection, and fraud detection, are the example of Text-based applications.

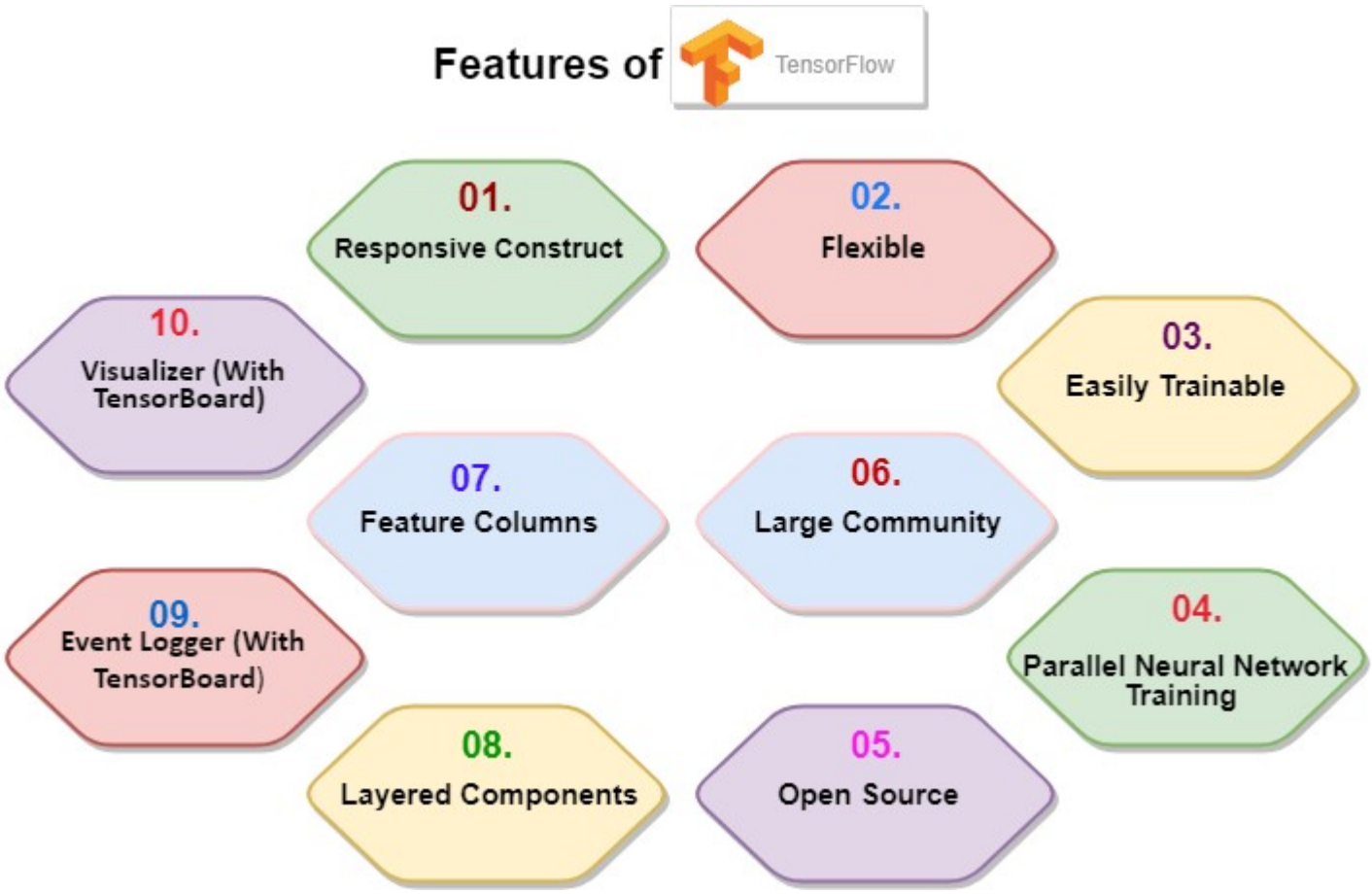
For example, Google Translate supports over 100 languages.

Some **companies** who are *currently using TensorFlow* are Google, AirBnb, eBay, Intel, DropBox, Deep Mind, Airbus, CEVA, Snapchat, SAP, Uber, Twitter, Coca-Cola, and IBM.

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

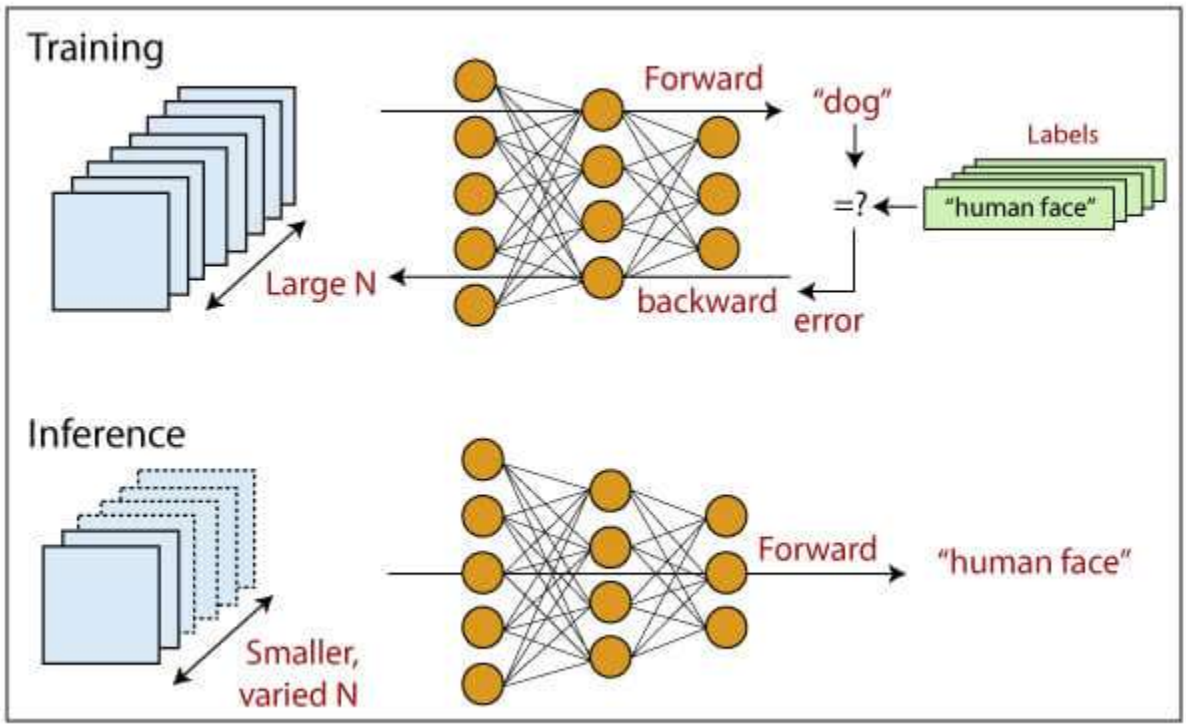
It is one of the essential TensorFlow Features according to its operability. It has modularity and parts of it which we want to make standalone.

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.

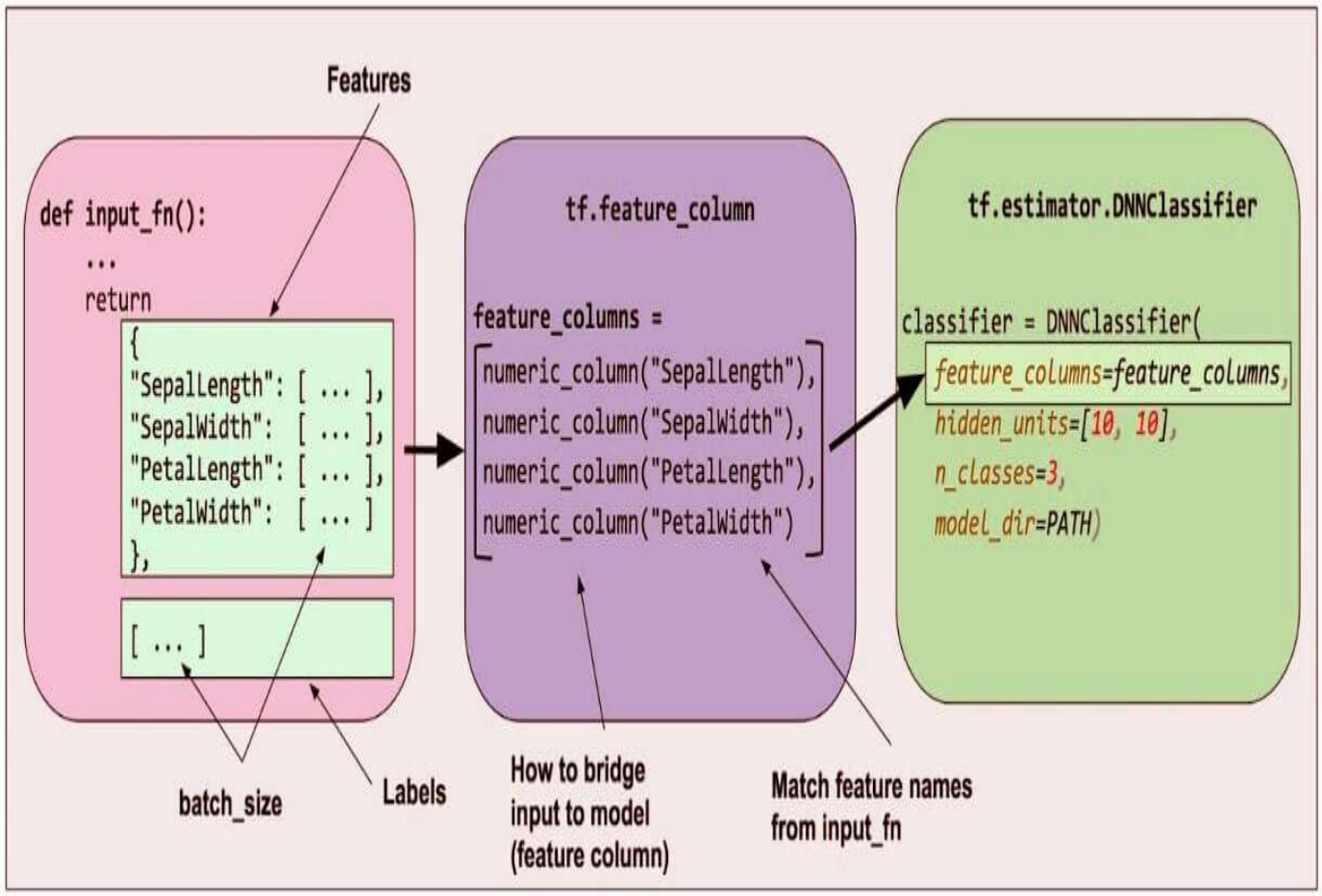
6. Open Source

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.



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.