TENSORFLOW

TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. TensorFlow was developed by Google Brain team for internal Google use in research and production. It can be used in a wide variety of programming languages, most notably python, as well as Javascript, C++ and java. It is available on 64-bit Linux, macOS, windows, and mobile computing platforms including Android and iOS. TensorFlow serves as the core platform and library for machine learning. TensorFlow's APIs use Keras to allow users to make their own machine learning models.

Graphs

- TensorFlow makes use of a graph framework. The graph gathers and computations done during the training.
- It was done to run on multiple CPUs or GPUs and even mobile operating
- The portability of the graph allows to preserve the computations for graph can be saved to be executed in the feature.
- All the computations in the graph are done by connecting tensors together. A tensor has a node and an edge.

Nodes :-

Each operation in the graph is represented by a node. Nodes are responsible for performing computations on data. Examples of nodes include operations like addition, multiplication, and matrix multiplication.

Edges :-

The edges of the graph represent the tensors (data) that flow between the nodes. The graph defines the sequence of operations that will be executed.

Tensor

A Tensor is the fundamental data structure in TensorFlow. It is a multi-dimensional array, similar to a NumPy array, but with added features that make it suitable for deep learning. Tensors are the "data" that flows through computational graph. Every Tensor has 3 key attributes.

1. Rank

The rank of a tensor is its number of dimensions.

- A scalar (a single number) has a rank of 0.
- A vector (a 1D array) has a rank of 1.
- A matrix (a 2D array) has a rank of 2.
- A 3D array has a rank of 3 and soon.

eg: tf.constant(42) ⟶ rank-0 tensor
tf.constant([1,2,3]) → rank-1 tensor.

2. Shape

The shape of a tensor is a tuple of integers that specifies the size of the tensor along each dimension.

- A scalar has an empty shape ().
- A vector with 5 elements has a shape of (5,).
- A matrix with 3 rows and 4 columns has a shape of (3,4).

eg: tf.constant([[1,2,3],[4,5,6]] has a shape of (2,3).

3. Type

The type of a tensor specifies the data type of its elements, such as float32, int32, uint8, etc. TensorFlow supports a wide range of data types.

eg: tf.constant(1.0) has a default type of float32.
tf.constant(1) has a default type of int32.

Building Neural Networks with TensorFlow

It is a straightforward process using the Keras API, which is now its official high level interface.

The basic workflow :-

1. Import Libraries : Start by importing tensorflow and its keras module.
2. Load and Prepare Data : Load your dataset and perform any necessary preprocessing, like normalizing pixel values for image data.
3. Define the model : Use keras.Sequential to build a model layer by layer. For eg: you can flatten input data with layers.Flatten, add hidden layers with layers.Dense, and use activation functions like relu or Softmax.
4. Compile the model : Configure the training process by specifying an optimizer (eg: 'adam'), a loss function (eg: 'sparse_categorical_crossentropy'), and metrics (eg: 'accuracy').
5. Train the Model : Use the model.fit() method to train the model on your data for a specified number of epochs.
6. Evaluate and Predict : Use model.evaluate() to check the model's performance on a test set and model.predict() to make predictions on new data.

Introduction to Keras

Keras is a high level user friendly API for building and training deep learning models. It was originally developed as a seperate project but has since been fully integrated into TensorFlow, making it the standard and most recommended way to build models in TensorFlow.

Keras is known for its simplicity, modularity, and extensibility. It allows developers to quickly prototype and build models without getting bogged down in low level details.

Core Concepts

1. Models : A model is the central Keras object that holds and organizes your layers. The two most common ways to build a model are Sequential API and Functional API.

2. Layers : Layers are the fundamental building blocks of a keras model. They are computational units that process input data and produce output. Keras offers a vast library of prebuilt layers for various tasks:
   - Dense : A standard, fully connected neural network
   - Conv2D, MaxPooling2D : Layers for building convolutional neural networks which are commonly used for image processing.
   - LSTM, GRU : Recurrent layers for handling sequential data like text or time series.
   - Flatten, Dropout : Utility layers for reshaping data or regularizing the model to prevent overfitting.

3. Compilation and Training : Keras simplifies the process of training a model with two key methods.
   (i) model.compile() :- This method configures the model for training. You specify 3 crucial components such as optimizer, loss function and metrics.
   (ii) model.fit() :- This is where the training happens. You provide the training data, labels and training parameters such as number of epochs and batch_size. Keras handles the complex back-propagation algorithm and weight updates automatically.