

# TensorFlow

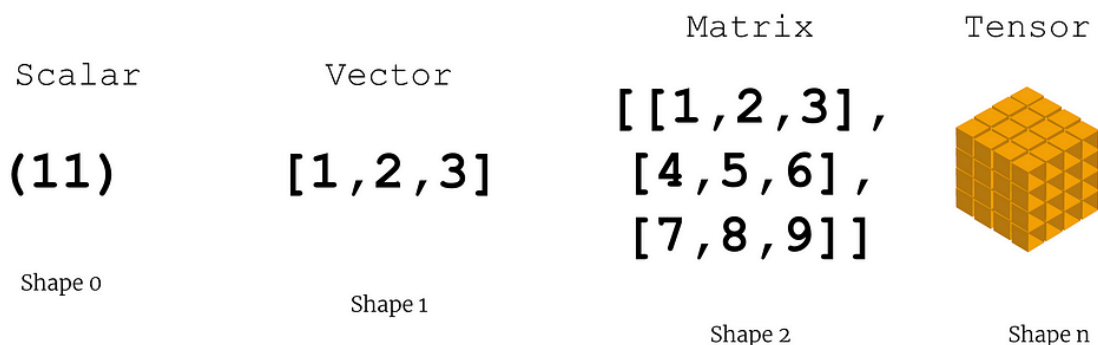
TensorFlow is a library that helps engineers build and train deep learning models.

We can use TensorFlow to train simple to complex neural networks using large sets of data.

TensorFlow enables us to quickly and easily build powerful AI models with high accuracy and performance.

## What is a Tensor?

A simple explanation would be that a tensor is a multi-dimensional array.



Scalar, Vector, Matrix and Tensor

A scalar is a single number. A vector is an array of numbers. A matrix is a 2-dimensional array. A tensor is an n-dimensional array.

In TensorFlow, everything can be considered a tensor including a scalar. A scalar would be a tensor of dimension 0, a vector of dimension 1, and a matrix of dimension 2.

## What is TensorFlow?

TensorFlow is an open-source software library for building neural networks. Google Brain team was the one who built it and it is the most popular deep learning library in the market today.

You can use TensorFlow to build AI models including image and speech recognition, natural language processing, and predictive modeling.

## TensorFlow and Keras

TensorFlow has a high-level API called Keras. Keras was a standalone project which is now available within the TensorFlow library. Keras makes it easy to define and train models while TensorFlow provides more control over the computation.

TensorFlow supports a wide range of hardware, including CPUs, GPUs, and TPUs. TPUs are Tensor processing Units, built specifically to work with Tensors and TensorFlow.

We can also run TensorFlow on mobile devices and IoT devices using TensorFlow Lite.

## How to Build Tensors with TensorFlow

Importing TensorFlow and printing out the version.

```
import tensorflow as tf
print(tf.__version__)
```

OUTPUT:

2.9.2

**First create a scalar using tf.constant.**

```
scalar = tf.constant(7)
print(scalar)
print(scalar.ndim)
```

OUTPUT:

```
tf.Tensor(7, shape=(), dtype=int32)
0
```

**Create a vector**

```
vector = tf.constant([10,10])
print(vector)
print(vector.ndim)
```

OUTPUT:

```
tf.Tensor([10 10], shape=(2,), dtype=int32)
1
```

**Creating a Matrix**

```
matrix = tf.constant([
    [10,11],
    [12,13]
])
print(matrix)
print(matrix.ndim)
```

OUTPUT:

```
tf.Tensor(
[[10 11]
 [12 13]], shape=(2, 2), dtype=int32)
2
```

**Tensors have a default datatype of int32.**

## Simple calculations using the tensor

```
basic_tensor = tf.constant([[10,11],[12,13]])
print(basic_tensor)
```

OUTPUT:

```
tf.Tensor(
[[10 11]
 [12 13]], shape=(2, 2), dtype=int32)
```

We can add, subtract, multiply, and divide every value in a tensor using the basic operators.

```
print(basic_tensor + 10)
print(basic_tensor - 10)
print(basic_tensor * 10)
print(basic_tensor / 10)
```

OUTPUT:

```
tf.Tensor(
[[20 21]
 [22 23]], shape=(2, 2), dtype=int32)
tf.Tensor(
[[0 1]
 [2 3]], shape=(2, 2), dtype=int32)
tf.Tensor(
[[100 110]
 [120 130]], shape=(2, 2), dtype=int32)
tf.Tensor(
[[1.  1.1]
 [1.2 1.3]], shape=(2, 2), dtype=float64)
```

## Matrix Multiplication

```
tensor_011 = tf.constant([[2,2],[4,4]])
tensor_012 = tf.constant([[2,3],[4,5]])

print(tf.matmul(tensor_011,tensor_012))
```

OUTPUT:

```
tf.Tensor(
[[12 16]
 [24 32]], shape=(2, 2), dtype=int32)
```

## To find the minimum and maximum values

```
tensor_013 = tf.constant([
    [1,2,3],
    [4,5,6],
    [7,8,9]
],dtype='float32')
print(tf.reduce_min(tensor_013))
print(tf.reduce_max(tensor_013))
print(tf.reduce_sum(tensor_013))
```

OUTPUT:

```
tf.Tensor(1.0, shape=(), dtype=float32)
tf.Tensor(9.0, shape=(), dtype=float32)
tf.Tensor(45.0, shape=(), dtype=float32)
```

## Find the square, square root, and log of each value in a tensor.

```
print(tf.sqrt(tensor_013))
print(tf.square(tensor_013))
print(tf.math.log(tensor_013))
```

OUTPUT:

```
tf.Tensor(  
[[1.          1.4142135 1.7320508]  
 [2.          2.236068  2.4494898]  
 [2.6457512 2.828427  3.          ]], shape=(3, 3), dtype=float32)  
tf.Tensor(  
[[ 1.  4.  9.]  
 [16. 25. 36.]  
 [49. 64. 81.]], shape=(3, 3), dtype=float32)  
tf.Tensor(  
[[0.          0.6931472 1.0986123]  
 [1.3862944 1.609438  1.7917595]  
 [1.9459102 2.0794415 2.1972246]], shape=(3, 3), dtype=float32)
```

## Conclusion

Tensorflow is a powerful library to build deep-learning models. It has all the tools we need to construct neural networks to solve problems like image classification, sentiment analysis, stock market predictions, etc.