**TensorFlow Operations:**

**Mathematical**

**Mathematical Operations with Tensors**

TensorFlow provides a plethora of mathematical operations for manipulating tensors. The numerical operations include addition, subtraction, multiplication, division, and more. Here’s an overview of some common mathematical operations along with their implementations:

Here, [tf.constant()](https://www.geeksforgeeks.org/python-tensorflow-constant/" \t "_blank) is used for defining tensors.

* [tf.add](https://www.geeksforgeeks.org/tensorflow-tf-add-function/) is used to perform element-wise addition on tensors a and b.
* [tf.subtract](https://www.geeksforgeeks.org/tensorflow-tf-sub-function/) is used for element-wise subtraction.
* [tf.multiply](https://www.geeksforgeeks.org/tensorflow-tf-mul-function/) performs element-wise multiplication.
* [tf.divide](https://www.geeksforgeeks.org/tensorflow-js-tf-div-function/) is used for element-wise division.

Python3

**import** **tensorflow** **as** **tf**

a = tf.constant([3, 3, 3])

b = tf.constant([2, 2, 2]) *# Define tensors*

sum\_result = tf.add(a, b) *# Addition*

diff\_result = tf.subtract(a, b) *# Subtraction*

quot\_result = tf.divide(a, b) *# Division*

prod\_result = tf.multiply(a, b) *# Multiplication*

print("Sum of Tensors",sum\_result)

print("Difference of tensors",diff\_result)

print("Quotient of Tensor",quot\_result)

print("Product of tensors ",prod\_result)

**Output:**

Sum of Tensors tf.Tensor([5 5 5], shape=(3,), dtype=int32)

Difference of tensors tf.Tensor([1 1 1], shape=(3,), dtype=int32)

Quotient of Tensor tf.Tensor([1.5 1.5 1.5], shape=(3,), dtype=float64)

Product of tensors tf.Tensor([6 6 6], shape=(3,), dtype=int32)

**Reduction**

**Aggregations and Statistics Operations with Tensors**

TensorFlow provides several aggregation functions for statistical analysis using**.reduce()**. Reduction operations in TensorFlow involve performing operations across specific axes of a tensor, reducing the dimensionality of the tensor in the process.

For implementation, the below code demonstrates how to calculate the mean along columns. The axis parameter specifies the axis along which the operations are performed.

* [tf.reduce\_mean](https://www.geeksforgeeks.org/python-tensorflow-math-reduce_mean/) is used to compute the mean along a specified axis (axis=0 means column-wise).

Python3

**import** **tensorflow** **as** **tf**

x = tf.constant([[1, 2, 3], [4, 5, 6]])

mean\_result = tf.reduce\_mean(x, axis=0)

print(mean\_result)

**Output:**

tf.Tensor([2 3 4], shape=(3,), dtype=int32)

**Logical AND and OR operations**

**tf.reduce\_all and [tf.reduce\_any:](https://www.geeksforgeeks.org/python-tensorflow-math-reduce_any/" \t "_blank)** Compute the logical AND and OR operations, respectively, across elements along a specified axis or axes.

Python3

**import** **tensorflow** **as** **tf**

x = tf.constant([[**True**, **False**, **True**], [**False**, **True**, **True**]])

all\_result = tf.reduce\_all(x, axis=1)

any\_result = tf.reduce\_any(x, axis=0)

print(all\_result)

print(any\_result)

**Output:**

tf.Tensor([False False], shape=(2,), dtype=bool)

tf.Tensor([ True True True], shape=(3,), dtype=bool)

**log-space operations**

[**tf.math.reduce\_logsumexp**](https://www.geeksforgeeks.org/python-tensorflow-math-reduce_logsumexp/)**:** Computes the logarithm of the sum of exponentials of elements along a specified axis or axes. This is useful for numerical stability in log-space operations.

Python3

**import** **tensorflow** **as** **tf**

x = tf.constant([[1.0, 2.0, 3.0], [4.0, 5.0, 6.0]])

logsumexp\_result = tf.math.reduce\_logsumexp(x, axis=1)

print(logsumexp\_result)

**Output:**

tf.Tensor([3.407606 6.407606], shape=(2,), dtype=float32)

**Matrix operations**

<https://www.tensorflow.org/api_docs/python/tf/linalg/matmul>

Matrix multiplication is probably is mostly used operation in machine learning, becase all images, sounds, etc are represented in matrixes.

There there are 2 types of multiplication:

**Element-wise multiplication : tf.multiply**

Element-wise multiplication in TensorFlow is performed using two tensors with identical shapes. This is because the operation multiplies elements in corresponding positions in the two tensors. An example of an element-wise multiplication, denoted by the **⊙** symbol, is shown below:

A number of numbers and symbols

Description automatically generated with medium confidence

**import** tensorflow **as** tf

A1 = tf.constant([1, 2, 3, 4])

B1 = tf.constant([3, 4, 5, 5])

C1 = tf.multiply(A1, B1)

*# C1 = <tf.Tensor: id=2, shape=(4,), dtype=int32, numpy=array([ 3, 8, 15, 20], dtype=int32)>*

**Matrix multiplication : tf.matmul**

**import** tensorflow **as** tf

A1 = tf.constant([[2, 24], [2, 26], [2, 57]])

B1 = tf.constant([[1000], [150]])

C1 = tf.matmul(A1, B1)

*# C1 = <tf.Tensor: id=5, shape=(3, 1), dtype=int32, numpy=*

array([[ 5600],

[ 5900],

[10550]], dtype=int32)>

**Matrix add : tf.add**

Note, that matrixes should be exactly the same shape

**import** tensorflow **as** tf

A1 = tf.constant([1,2,3])

B1 = tf.constant([1,2,3])

C1 = tf.add(A1, B1)

*# C1 = <tf.Tensor: id=20, shape=(3,), dtype=int32, numpy=array([2, 4, 6], dtype=int32)>*

**Matrix sum by dimension : tf.reduce\_sum()**

This operator just sum all elements of the matrix or specific row or column

**import** tensorflow **as** tf

A1 = tf.Variable([[1,2,3],[3,2,1], [3,3,3]])

B2 = tf.reduce\_sum(A1)

*# B2 <tf.Tensor: id=34, shape=(), dtype=int32, numpy=21>*

B3 = tf.reduce\_sum(A1, 0)

*#B3 <tf.Tensor: id=37, shape=(3,), dtype=int32, numpy=array([7, 7, 7], dtype=int32)>*

B4 = tf.reduce\_sum(A1, 1)

*#B4 <tf.Tensor: id=46, shape=(3,), dtype=int32, numpy=array([6, 6, 9], dtype=int32)>*

Ref: <https://medium.com/analytics-vidhya/matrix-operations-using-tensorflow-61a6666ded8f>

**Data Manipulation Operations:**

<https://www.tensorflow.org/tutorials/customization/basics>

In TensorFlow, data manipulation is a crucial step in preparing your data for machine learning models. TensorFlow provides a variety of functions and tools to handle, transform, and preprocess data. Here’s a rundown of common data manipulation operations you might use: A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

**A black and white text on a white background

Description automatically generated**

These operations provide a broad range of functionalities for manipulating data in TensorFlow. The specifics of how you use these functions will depend on your particular use case and the nature of your data.

**Activation Functions, Convolution Operations, and Recurrent Operation**

<https://subscription.packtpub.com/book/data/9781786462169/1/ch01lvl1sec15/implementing-activation-functions>

<https://www.tensorflow.org/api_docs/python/tf/nn/convolution>

**Loss Functions and Gradient Operations**

**Image Operations**

**TensorFlow Estimators:**

Predefined Estimators including linear regression, logistic regression, deep neural networks (DNNs), and gradient boosting models (GBMs).

Using train() and evaluate() with an estimator

**Tf.data**  
Data loading including built-in datasets such as MNIST and CIFAR-10

Transformation and Preprocessing with map(), filter(), batch(), and etc.