**EXPERIMENT 2**

**Aim: Create tensors and apply split and merge operations and statistics operations**

**Description:**

In TensorFlow, tensors are the core data structures used to represent and manipulate multi-dimensional arrays or numerical data. A tensor is a generalized representation of a matrix, extending to higher dimensions. TensorFlow provides several ways to create tensors:

### 1. ****From Python Objects****

You can create tensors directly from Python lists, tuples, or NumPy arrays using tf.constant or tf.convert\_to\_tensor.

tf.constant : Creates a constant tensor with a fixed value.

import tensorflow as tf

# Create a tensor from a list

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

Output:

<tf.Tensor: shape=(2, 2), dtype=int32, numpy=array([[1, 2], [3, 4]], dtype=int32)>

tf.convert\_to\_tensor : Converts an existing data structure (like NumPy arrays) into a tensor.

import numpy as np

numpy\_array = np.array([[5.0, 6.0], [7.0, 8.0]])

tensor = tf.convert\_to\_tensor(numpy\_array)print(tensor)

### 2. ****Using TensorFlow Functions****

TensorFlow provides a range of functions for creating tensors with specific values or patterns.

**Zeros and Ones**

zeros = tf.zeros([3, 3]) # Tensor of shape (3, 3) filled with 0s

ones = tf.ones([2, 4]) # Tensor of shape (2, 4) filled with 1s

**Random Values**

* + tf.random.uniform: Generates random values uniformly distributed in a range.
  + tf.random.normal: Generates random values from a normal distribution.

uniform = tf.random.uniform([3, 3], minval=0, maxval=10) # Random values between 0 and 10

normal = tf.random.normal([3, 3], mean=0, stddev=1) # Random values with mean=0 and stddev=1

**Identity Tensor**

identity = tf.eye(4) # 4x4 Identity matrix

**Range and Sequence**

tf.range: Creates a sequence of values.

sequence = tf.range(start=1, limit=10, delta=2) # Values: 1, 3, 5, 7, 9

### 3. ****From Existing Tensors****

**Reshape**: Change the shape of a tensor without altering its data.

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

reshaped = tf.reshape(original, [2, 3]) # Shape: 2 rows, 3 columns

**Broadcasting**: Extend the dimensions of a tensor to make it compatible for operations with other tensors.

scalar = tf.constant(5)

broadcasted = tf.broadcast\_to(scalar, [3, 3]) # 3x3 tensor filled with 5

### 4. ****Dynamic Tensors****

Tensors in TensorFlow are immutable, but operations can generate new tensors. TensorFlow also supports tf.Variable for tensors whose values can be updated.

variable\_tensor = tf.Variable([[1, 2], [3, 4]])

variable\_tensor.assign([[5, 6], [7, 8]]) # Update the values of the variable tensor

### 5. ****GPU/Device Placement****

Tensors are automatically assigned to CPU or GPU, but you can explicitly specify the device using TensorFlow's with statement:

with tf.device('/GPU:0'): # Create tensor on GPU if available

gpu\_tensor = tf.constant([1.0, 2.0, 3.0])

**Different Types of Operations Performed**

Tensor Creation:

* Creates a tensor with user-defined dimensions and initializes it with random values, which are then replaced by user inputs.
* Functions used: numpy.random.rand, float(input)

Tensor Splitting:

* Splits the tensor into smaller sub-tensors along a specified axis.
* Function used: numpy.split

Tensor Merging:

* Merges the previously split sub-tensors back together along the same axis.
* Function used: numpy.concatenate

Mean Calculation:

* Calculates the mean of the tensor elements.
* Function used: numpy.mean

Standard Deviation Calculation:

* Calculates the standard deviation of the tensor elements.
* Function used: numpy.std

Variance Calculation:

* Calculates the variance of the tensor elements.
* Function used: numpy.var

Finding Maximum and Minimum Values:

* Identifies the maximum and minimum values within the tensor.
* Functions used: numpy.max, numpy.min

Tensor Reshaping:

* Reshapes the tensor into a 1D array.
* Function used: numpy.reshape

Tensor Transposition:

* Transposes the tensor, swapping its rows and columns.
* Function used: numpy.transpose

Dot Product:

* Performs a dot product between the tensor and its transpose.
* Function used: numpy.dot

| **Operation Type** | **Example Functions** |
| --- | --- |
| Arithmetic | tf.add, tf.multiply, tf.divide |
| Matrix | tf.matmul, tf.transpose, tf.linalg.inv |
| Aggregation | tf.reduce\_sum, tf.reduce\_mean |
| Logical | tf.greater, tf.equal, tf.logical\_and |
| Reshaping & Slicing | tf.reshape, tf.slice, tf.gather |
| Broadcasting | Implicit in operations |
| Stacking & Concatenation | tf.stack, tf.concat |
| Casting | tf.cast |