

# Maximum element of a Tensor

Maximum element

```
x = tf.constant([[9,2,10,4],[5,6,7,8]])  
print(tf.reduce_max(x))
```

```
tf.Tensor(10, shape=(), dtype=int32)
```

```
[ 9, 2, 10, 4]  
[ 5, 6, 7 , 8]
```

Index of the Maximum element

```
x = tf.constant([[9,2,10,4],[5,6,7,8]])  
print(tf.math.argmax(x))
```

```
tf.Tensor([0 1 0 1], shape=(4,), dtype=int64)
```

```
[ 9, 2, 10, 4]  
[ 5, 6, 7 , 8]
```

```
x = tf.constant([[2, 20, 30, 3, 6], [3, 11, 16, 1, 8],  
                [14, 45, 23, 5, 27]])  
print(tf.math.argmax(x))
```

```
tf.Tensor([2 2 0 2 2], shape=(5,), dtype=int64)
```

```
[ 2, 20, 30, 3, 6 ]  
[ 3, 11, 16, 1, 8 ]  
[ 14, 45, 23, 5, 27 ]
```

# Minimum element of a Tensor

## Minimum element

```
x = tf.constant([[9,2,10,4],[5,6,7,8]])
print(tf.reduce_min(x))

tf.Tensor(2, shape=(), dtype=int32)
```

```
[ 9, 2, 10, 4]
[ 5, 6, 7, 8]
```

## Index of the Minimum element

```
x = tf.constant([[9,2,10,4],[5,6,7,8]])
print(tf.math.argmax(x))

tf.Tensor([1 0 1 0], shape=(4,), dtype=int64)
```

```
[ 9, 2, 10, 4]
[ 5, 6, 7, 8]
```

```
x = tf.constant([[2, 20, 30, 3, 6], [3, 11, 16, 1, 8],
                [14, 45, 23, 5, 27]])
print(x)
print(tf.math.argmax(x))
```

```
tf.Tensor(
[[ 2 20 30  3  6]
 [ 3 11 16  1  8]
 [14 45 23  5 27]], shape=(3, 5), dtype=int32)
tf.Tensor([0 1 1 1 0], shape=(5,), dtype=int64)
```

```
[ 2, 20, 30, 3, 6 ]
[ 3, 11, 16, 1, 8 ]
[14, 45, 23, 5, 27]
```

# Minimum/Maximum of Two Tensors

## Minimum

```
x = tf.constant([0., 0., 0., 0.])  
y = tf.constant([-5., -2., 0., 3.])  
tf.math.minimum(x, y)
```

```
<tf.Tensor: shape=(4,), dtype=float32, numpy=array([-5., -2., 0., 0.], dtype=float32)>
```

## Maximum

```
x = tf.constant([0., 0., 0., 0.])  
y = tf.constant([-5., -2., 0., 3.])  
tf.math.maximum(x, y)
```

```
<tf.Tensor: shape=(4,), dtype=float32, numpy=array([0., 0., 0., 3.], dtype=float32)>
```

# Concatenation of Two Tensors

## Along 0-Axis

```
x = [[1, 2, 3], [4, 5, 6]]  
y = [[7, 8, 9], [10, 11, 12]]  
tf.concat([x, y], 0)
```

```
<tf.Tensor: shape=(4, 3), dtype=int32, numpy=  
array([[ 1,  2,  3],  
       [ 4,  5,  6],  
       [ 7,  8,  9],  
       [10, 11, 12]])>
```

## Along 1-Axis

```
x = [[1, 2, 3], [4, 5, 6]]  
y = [[7, 8, 9], [10, 11, 12]]  
tf.concat([x, y], 1)
```

```
<tf.Tensor: shape=(2, 6), dtype=int32, numpy=  
array([[ 1,  2,  3,  7,  8,  9],  
       [ 4,  5,  6, 10, 11, 12]])>
```

---

# Modifying the value of a Tensor

**Not a simple operation.**

**Possible for Variable type. But, for such operation, we would prefer to use `numpy` library.**

# Matrix Multiplication

```
import tensorflow as tf

A1 = tf.constant([[1, 2, 3, 4]])
B1 = tf.constant([[3], [4], [5], [5]])
C1 = tf.multiply(A1, B1)
tf.print(C1)
```

```
[[3 6 9 12]
 [4 8 12 16]
 [5 10 15 20]
 [5 10 15 20]]
```

```
import tensorflow as tf

A1 = tf.constant([[1, 2, 3, 4]])
B1 = tf.constant([[3], [4], [5], [5]])
C1 = tf.matmul(A1, B1)
tf.print(C1)
```

```
[[46]]
```