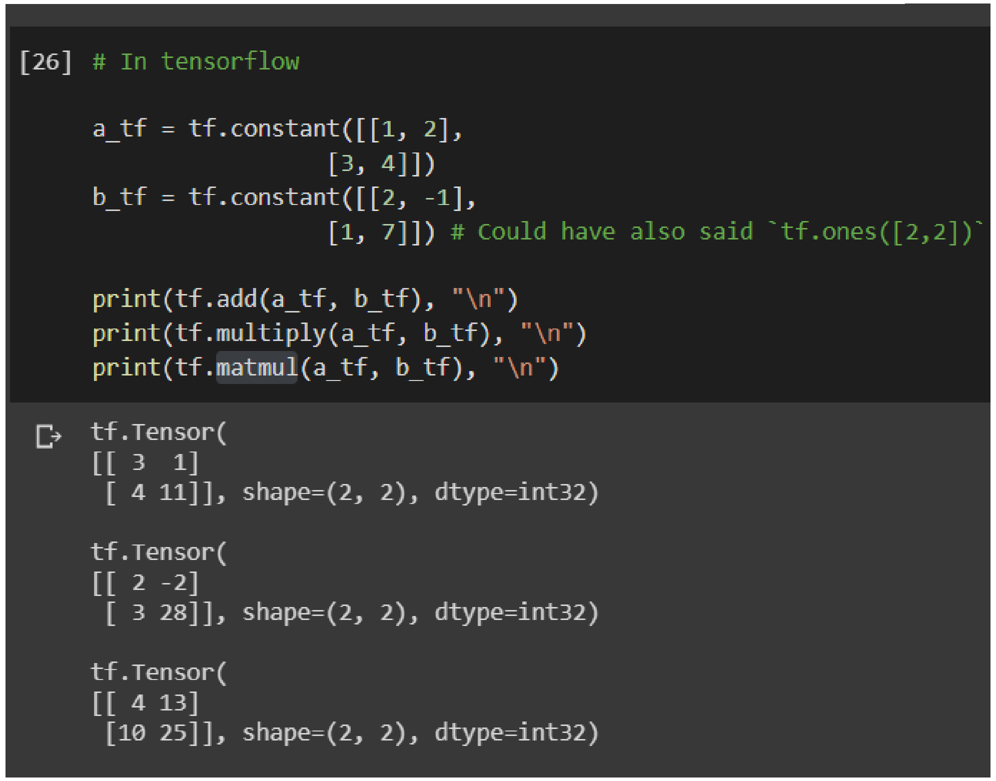
TensorFlow VS NumPy

- By Vikraant Pai, Mtech Data Science, D005, NMIMS

Link to the notebook:
https://github.com/vikpy/AISem3/blob/master/HW/HomeWork2_Tensor_Vector_Binary_Tree.ipynb

Operations:

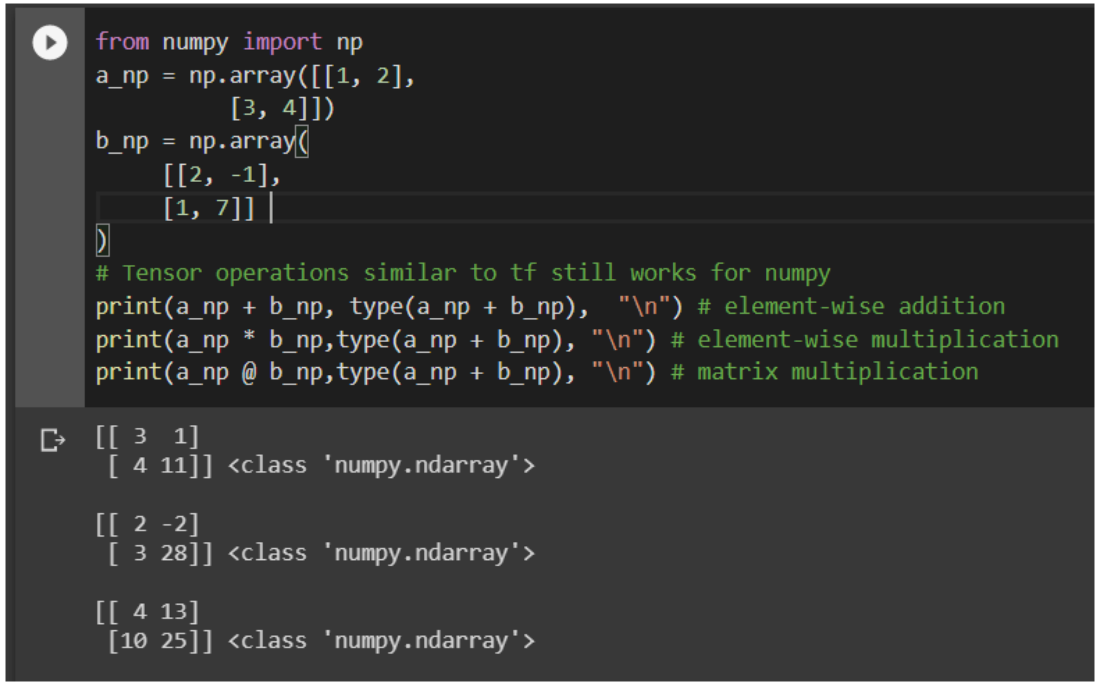
TensorFlow



```
[26] # In tensorflow

     a_tf = tf.constant([[1, 2],
                          [3, 4]])
     b_tf = tf.constant([[2, -1],
                          [1, 7]]) # Could have also said `tf.ones([2,2])`

     print(tf.add(a_tf, b_tf), "\n")
     print(tf.multiply(a_tf, b_tf), "\n")
     print(tf.matmul(a_tf, b_tf), "\n")
```

```
tf.Tensor(
[[ 3  1]
 [ 4 11]], shape=(2, 2), dtype=int32)

tf.Tensor(
[[ 2 -2]
 [ 3 28]], shape=(2, 2), dtype=int32)

tf.Tensor(
[[ 4 13]
 [10 25]], shape=(2, 2), dtype=int32)
```

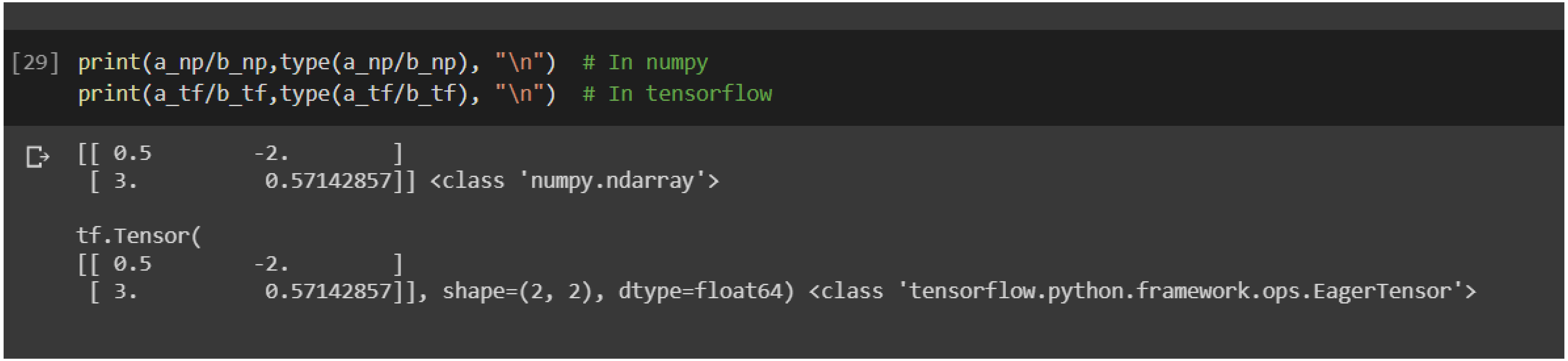
NumPy



```
from numpy import np
a_np = np.array([[1, 2],
          [3, 4]])
b_np = np.array(
      [[2, -1],
      [1, 7]]
)
# Tensor operations similar to tf still works for numpy
print(a_np + b_np, type(a_np + b_np),  "\n") # element-wise addition
print(a_np * b_np,type(a_np + b_np), "\n") # element-wise multiplication
print(a_np @ b_np,type(a_np + b_np), "\n") # matrix multiplication
```

```
[[ 3  1]
 [ 4 11]] <class 'numpy.ndarray'>

[[ 2 -2]
 [ 3 28]] <class 'numpy.ndarray'>

[[ 4 13]
 [10 25]] <class 'numpy.ndarray'>
```

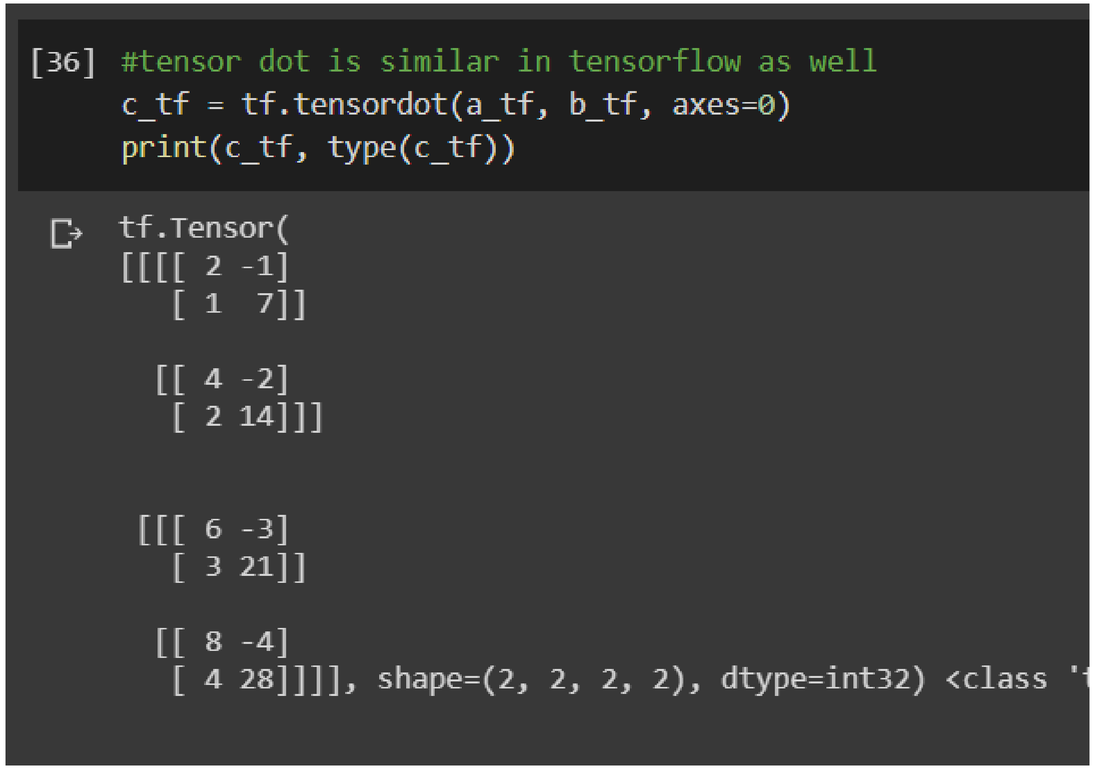
Tensor division operation:



```
[29] print(a_np/b_np,type(a_np/b_np), "\n")  # In numpy
     print(a_tf/b_tf,type(a_tf/b_tf), "\n")  # In tensorflow
```

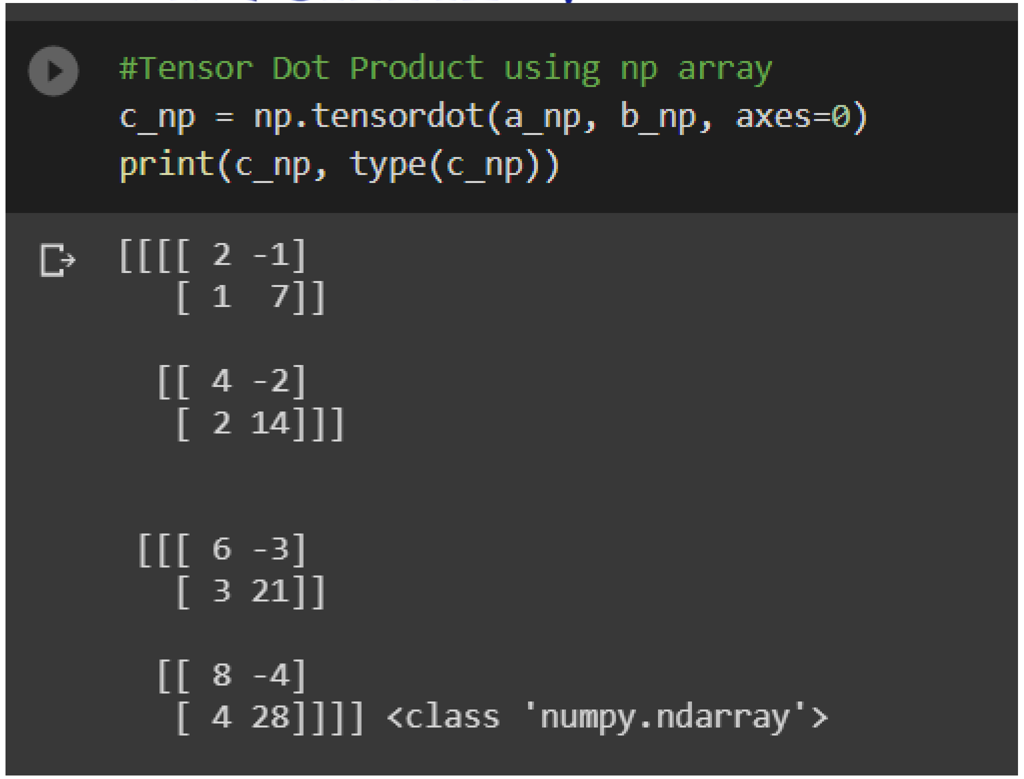
```
[[ 0.5         -2.        ]
 [ 3.          0.57142857]] <class 'numpy.ndarray'>

tf.Tensor(
[[ 0.5         -2.        ]
 [ 3.          0.57142857]], shape=(2, 2), dtype=float64) <class 'tensorflow.python.framework.ops.EagerTensor'>
```

Tensor dot Operation:



```
[36] #tensor dot is similar in tensorflow as well
     c_tf = tf.tensordot(a_tf, b_tf, axes=0)
     print(c_tf, type(c_tf))
```

```
tf.Tensor(
[[[[ 2 -1]
   [ 1  7]]

  [[ 4 -2]
   [ 2 14]]]


 [[[ 6 -3]
   [ 3 21]]

  [[ 8 -4]
   [ 4 28]]]], shape=(2, 2, 2, 2), dtype=int32) <class '
```



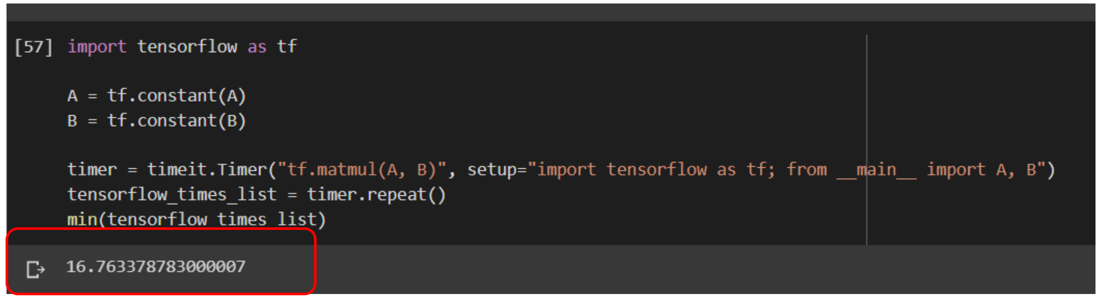
```
#Tensor Dot Product using np array
c_np = np.tensordot(a_np, b_np, axes=0)
print(c_np, type(c_np))
```

```
[[[[ 2 -1]
   [ 1  7]]

  [[ 4 -2]
   [ 2 14]]]


 [[[ 6 -3]
   [ 3 21]]

  [[ 8 -4]
   [ 4 28]]]] <class 'numpy.ndarray'>
```

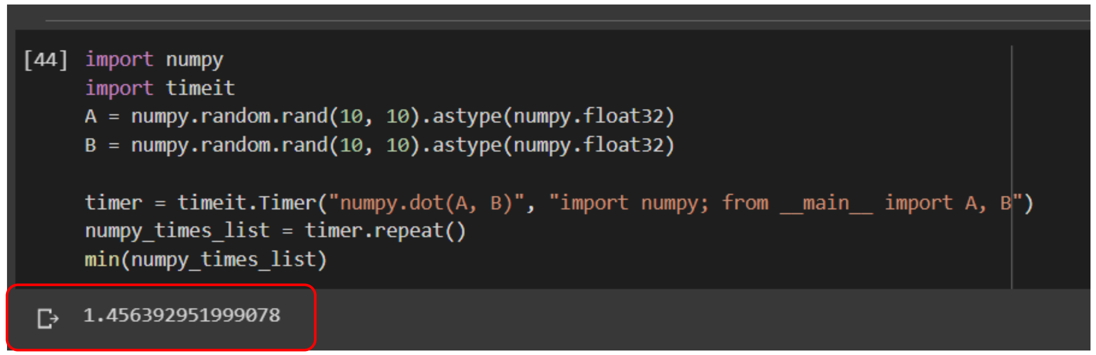
Speed comparison Operation:

TensorFlow

NumPy



```
[57] import tensorflow as tf

     A = tf.constant(A)
     B = tf.constant(B)

     timer = timeit.Timer("tf.matmul(A, B)", setup="import tensorflow as tf; from __main__ import A, B")
     tensorflow_times_list = timer.repeat()
     min(tensorflow_times_list)

16.763378783000007
```



```
[44] import numpy
     import timeit
     A = numpy.random.rand(10, 10).astype(numpy.float32)
     B = numpy.random.rand(10, 10).astype(numpy.float32)

     timer = timeit.Timer("numpy.dot(A, B)", "import numpy; from __main__ import A, B")
     numpy_times_list = timer.repeat()
     min(numpy_times_list)

1.456392951999078
```

Thank You!

Link to the notebook:
https://github.com/vikpy/AISem3/blob/master/HW/HomeWork2_Tensor_Vector_Binary_Tree.ipynb