# Debugging Notes:

&&&&& Traces for Addition Operation in tensorflow graph representation &&&&&:

\*\* math\_ops.py line 809

**def** binary\_op\_wrapper(x, y):

\*\* ops.py line 6018

name\_scope, context manager

def \_\_init\_\_( default name add, values: classList

\*\*math\_ops.py

**return** func(x, y, name=name) line 812

gen\_math\_ops.py -> def add(x,y,name)

\_, \_, \_op = \_op\_def\_lib.\_apply\_op\_helper( line 373

Op\_def\_library.py ->

g = ops.\_get\_graph\_from\_inputs(\_Flatten(keywords.values())) line 350

ops.py - > **if** get\_default\_graph().building\_function: line 5684

line 5687 op\_input\_list = tuple(op\_input\_list)

contextlib.py line 159

In op\_def\_library.py the def \_apply\_op\_helper is really large method,

In the end of it calls the

*# Add Op to graph*op = g.create\_op(op\_type\_name, inputs, output\_types, name=scope,  
 input\_types=input\_types, attrs=attr\_protos,  
 op\_def=op\_def)

&&&&& Traces for Constant Operation in tensorflow graph &&&&&

Constant\_op.py line 118

**def** constant\_v1(

value :3 , dtype:int32, name: Const

Creates a constant tensor

**def** \_constant\_impl(

const\_tensor = g.create\_op(  
 **"Const"**, [], [dtype\_value.type],  
 attrs={**"value"**: tensor\_value,  
 **"dtype"**: dtype\_value},  
 name=name).outputs[0]

So here in constant case call the create\_op function of ops.py.

Q: What is Tensor in TensorFlow?

Answer: Tensors are the Edges of the graph, which are multidimensional Arrays (Tensors).

So either they can have a dimension 0, meaning scalar value.

Dimension 1 , meaning a line or vector.

Dimension 2 , matrix Etc.

Tensor is really useful when dealing with images, so we can represent it with a 2-dimensional structure as a matrix. But since images have also colours we need another dimension, so 3 dimension should be used here.