**import** tensorflow **as** tf  
**import** numpy **as** np  
**from** PIL **import** Image  
**from** tensorflow.examples.tutorials.mnist **import** input\_data

* Basic operation: add, subtract, multiply and tensor run

X1 = tf.constant(5) % create constant

X2 = tf.constant(6)

Add = tf.add(x1,x2) % create add graph flow

Mult = tf.multiply(x1,x2) % create multiply graph flow

Sub = tf.subtract(x1,x2) % create subtract graph flow

Div = tf.divid(x1,x2) % create divide graph flow

X1

dividee

X2

DIV

The Add, mult, sub, are output of the operation. However, it is not the result yet(it is a tf object call “tensor” )

We have to run the session to run the tensor to get the number.

sess = tf.Session()  
*# Perform computations:*output\_add = sess.run(add) % ouput\_add = 5+6=11  
  
*# Close*sess.close() % don’t forget to close the session

we just need to run the tensor that we want to know, if the tensor input is other tensor, it will trace back and run the input tensor automatically.

addition

DIV

additione

dividee

* The tensor can be a array or matrix

tensor1 = tf.Variable([[1, 2, 3], [4, 5, 6]], tf.float64)  
tensor2 = tf.Variable([[4, 5, 6], [1, 2, 3]], tf.float64)

add = tf.add(tensor1, tensor2)  
sub = tf.subtract(tensor1, tensor2)  
mul = tf.multiply(tensor1, tensor2)  
div = tf.truediv(tensor1, tensor2)

init\_op = tf.global\_variables\_initializer()

with tf.Session as sess:

sess.run(init\_op)

print(sess.run(sub))

print(sess.run(mul))

print(sess.run(div))

print(sess.run(add))

* How to download the training picture

mnist = input\_data.read\_data\_sets(**"/tmp/data/"**, one\_hot=**True**)

tmp/data is the directory to save the file

* Create the space that we can put input in

x = tf.placeholder(tf.float32, [**3**,3]) # create a 3\*3 float matrix  
y = tf.matmul(x, x)  
  
**with** tf.Session() **as** sess:  
 rand\_array = np.random.rand(**3**,3)  
 print(rand\_array)  
 print(sess.run(y, feed\_dict={x: rand\_array})) *# put the rand\_array into x*

* Shuffle

tensor = tf.random\_shuffle(x) # it will shuffle the outter matrix

# ex: if input is 4\*3 matrix it will only shuffle the rows

* Slicing

var1 = tf.placeholder(tf.float32, shape=[3, 1, 2, 2])

tensor = tf.slice(var1, [0, 0, 0, 0], [2, 1, 1, 2])

*# Format: tf.slice(input\_var, start, size\_in\_each\_dim)*

# 'input' is [[[1, 1, 1], [2, 2, 2]],  
#             [[3, 3, 3], [4, 4, 4]],  
#             [[5, 5, 5], [6, 6, 6]]]  
tf.slice(input, [1, 0, 0], [1, 1, 3]) ==> [[[3, 3, 3]]]  
tf.slice(input, [1, 0, 0], [1, 2, 3]) ==> [[[3, 3, 3],  
                                            [4, 4, 4]]]  
tf.slice(input, [1, 0, 0], [2, 1, 3]) ==> [[[3, 3, 3]],  
                                           [[5, 5, 5]]]

* Reshape

tensor = tf.reshape(var1, [var1\_size]) # tf.reshape(input\_var, ideal\_size)

* Create a variable matrix

Var = tf.Variable()