1. 常量

**import** tensorflow **as** tf

*# Create TensorFlow object called hello\_constant*

hello\_constant = tf.constant('Hello World!')

**with** tf.Session() **as** sess:

*# Run the tf.constant operation in the session*

output = sess.run(hello\_constant)

print(output)

2.变量

x = tf.placeholder(tf.string)

**with** tf.Session() **as** sess:

output = sess.run(x, feed\_dict={x: 'Hello World'})

x = tf.placeholder(tf.string)

y = tf.placeholder(tf.int32)

z = tf.placeholder(tf.float32)

**with** tf.Session() **as** sess:

output = sess.run(x, feed\_dict={x: 'Test String', y: 123, z: 45.67})

3.tensorflow Math

x = tf.add(5, 2) *# 7*

x = tf.subtract(10, 4) *# 6*

y = tf.multiply(2, 5) *# 10*

类型转换

tf.subtract(tf.constant(2.0),tf.constant(1)) # Fails with ValueError: Tensor conversion requested dtype float32 for Tensor with dtype int32:

**tf.Variable()**

x = tf.Variable(5)

The **[tf.Variable](https://www.tensorflow.org/api_docs/python/tf/Variable" \t "_blank)** class creates a tensor with an initial value that can be modified, much like a normal Python variable. This tensor stores its state in the session, so you must initialize the state of the tensor manually. You'll use the **[tf.global\_variables\_initializer()](https://www.tensorflow.org/programmers_guide/variables" \t "_blank)** function to initialize the state of all the Variable tensors.

**Initialization**

init = tf.global\_variables\_initializer()

**with** tf.Session() **as** sess:

sess.run(init)

分类问题

机器学习不止分类问题，但是掌握了分类，学习检测或排序这些将变得非常简单；

Similarly, choosing weights from a normal distribution prevents any one weight from overwhelming other weights. You'll use the **[tf.truncated\_normal()](https://www.tensorflow.org/api_docs/python/tf/truncated_normal" \t "_blank)** function to generate random numbers from a normal distribution.

### tf.truncated\_normal()

### tf.truncated\_normal()

n\_features = 120

n\_labels = 5

weights = tf.Variable(tf.truncated\_normal((n\_features, n\_labels)))

The **[tf.truncated\_normal()](https://www.tensorflow.org/api_docs/python/tf/truncated_normal" \t "_blank)** function returns a tensor with random values from a normal distribution whose magnitude is no more than 2 standard deviations from the mean.

### tf.zeros()

n\_labels = 5

bias = tf.Variable(tf.zeros(n\_labels))

Since xW in xW + b is matrix multiplication, you have to use the **[tf.matmul()](https://www.tensorflow.org/api_docs/python/tf/matmul" \t "_blank)** function instead of **[tf.multiply()](https://www.tensorflow.org/api_docs/python/tf/multiply" \t "_blank)**. Don't forget that order matters in matrix multiplication, so tf.matmul(a,b) is not the same as tf.matmul(b,a).