Chapter 6 - Hello Tensor World

March 25, 2018

0.1 Hello, Tensor World!

```
In [1]: 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)
```

```
/home/supannee/tensorflow/lib/python3.5/site-packages/h5py/__init__.py:36: FutureWarning: Conver
  from ._conv import register_converters as _register_converters
```

```
In [2]: # To get rid of the above warnings
        import os
        os.environ["TF_CPP_MIN_LOG_LEVEL"]="3"

        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)
```

```
b'Hello World!'
```

0.1.1 Tensor

In TensorFlow, data isn't stored as integers, floats, or strings. These values are encapsulated in an object called a **tensor**. In this case, `hello_constant = tf.constant('Hello World!')`, `hello_constant` is a 0-dimensional string tensor.

A constant tensor, returned by `tf.constant()`, has a constant value, which is never changed.

```
In [3]: # A is a 0-dimensional int32 tensor
        A = tf.constant(1234)
        print(A.get_shape())

        # B is a 1-dimensional int32 tensor
        B = tf.constant([1, 2, 3])
        print(B.get_shape())

        # C is a 2-demensional int32 tensor
        C = tf.constant([[1,2,3],[4,5,6]])
        print(C.get_shape())

        # D is a 3-dimensional int32 tensor
        D = tf.constant([[[1,2,3,4],[1,2,3,4]],[[5,6,7,8],[5,6,7,8]]])
        print(D.get_shape())

        # E is a 4-dimensional int32 tensor
        E = tf.constant([[[[1,1],[2,2],[3,3]]],[[[4,4],[5,5],[6,6]]]]) # (2x1x3x2)
        print(E.get_shape())
```

```
()
(3,)
(2, 3)
(2, 2, 4)
(2, 1, 3, 2)
```

0.1.2 Session

TensorFlow's api is built around the idea of a computational graph. A *TensorFlow Session* is an environment for running a graph. The session is in charge of allocating the operations to GPU(s) and/or CPU(s).

```
# a session instance is created using tf.Session
with tf.Session as sess:
   # sess.run() evaluates the tensor and returns the results
   output = sess.run(hello_constant)
   print(output)
```