CHAPTER-9-TENSORFLOW

2. Is the statement a_val = a.eval(session=sess) equivalent to a_val = sess.run(a)?
Answer→
Yes, the statement a_val = a.eval(session=sess) is equivalent to a_val = sess.run(a).
They produce the same results

3. Is the statement a_val, b_val = a.eval(session=sess), b.eval(session=sess) equivalent to a_val, b_val = sess.run([a, b])?

Answer→No. and that is because of number of times graph runs. For expression one it runs twice and for second expression it runs once only. If no side effects are in both the cases they give us the same results else second statement will be faster than first.

5. If you create a graph g containing a variable w, then start two threads and open a session in each thread, both using the same graph g, will each session have its own copy of the variable w or will it be shared?

Answer→

In local TensorFlow, sessions manage variable values, so if you create a graph g containing a variable w, then start two threads and open a local session in each thread, both using the same graph g, then each session will have its own copy of the variable w. However, in distributed TensorFlow, variable values are stored in containers managed by the cluster, so if both sessions connect to the same cluster and use the same container, then they will share the same variable value for w.

6. When is a variable initialized? When is it destroyed?

Answer→
A variable initialized when you call its initializer and it ends when we end session. But in the case pf distributed TensorFlow live container on cluster and ending session won't destroy variable. You have to clear container to destroy variable.

7. What is the difference between a placeholder and a variable?

Answer→

Variables is an oprtaion of an assignment and you have to initialize it in order to run it. We can directly change variable name but it will keep the same value. It is typically used to hold model parameters but also for other purposes (e.g., to count the global training step).

On the other hand, Placeholders just hold the information about type and shape of the tensor they represent, but they have no value stored in it. We have to feed value and shape otherwise it results in exception and they are fed to TensorFlow during exception phase. We have to change model weights to change variables.

10. How can you set a variable to any value you want (during the execution phase)?

Answer→

While constructing the graph, we can specify a variable's initial value and it will be initialized later when you run the variable's initializer during the execution phase. If you want to change that variable's value to anything you want during the execution phase, then the simplest option is to create an assignment node (during the graph construction phase) using the tf.assign() function, passing the variable and a placeholder as parameters. During the execution phase, you can run the assignment operation and feed the variable's new value using the placeholder.

Here is a syntax to run samples.

```
import tensorflow as tf
x = tf.Variable(tf.random_uniform(shape=(), minval=0.0, maxval=1.0))
x_new_val = tf.placeholder(shape=(), dtype=tf.float32)
x_assign = tf.assign(x, x_new_val)
with tf.Session():
x.initializer.run() # random number is sampled *now*
print(x.eval()) # 0.646157 (some random number)
x_assign.eval(feed_dict={x_new_val: 5.0})
print(x.eval()) # 5.0
```