

Constants, Sequences, and Random Values

Note: Functions taking `Tensor` arguments can also take anything accepted by [tf.convert_to_tensor](#).

Constant Value Tensors

TensorFlow provides several operations that you can use to generate constants.

- `tf.zeros`
- `tf.zeros_like`
- `tf.ones`
- `tf.ones_like`
- `tf.fill`
- `tf.constant`

Sequences

- `tf.linspace`
- `tf.range`

Random Tensors

TensorFlow has several ops that create random tensors with different distributions. The random ops are stateful, and create new random values each time they are evaluated.

The `seed` keyword argument in these functions acts in conjunction with the graph-level random seed. Changing either the graph-level seed using `tf.set_random_seed` or the op-level seed will change the underlying seed of these operations. Setting neither graph-level nor op-level seed, results in a random seed for all operations. See [tf.set_random_seed](#) for details on the interaction between operation-level and graph-level random seeds.

Examples:

```
# Create a tensor of shape [2, 3] consisting of random normal values,
with mean
# -1 and standard deviation 4.
norm = tf.random_normal([2, 3], mean=-1, stddev=4)

# Shuffle the first dimension of a tensor
c = tf.constant([[1, 2], [3, 4], [5, 6]])
shuff = tf.random_shuffle(c)

# Each time we run these ops, different results are generated
sess = tf.Session()
print(sess.run(norm))
print(sess.run(norm))

# Set an op-level seed to generate repeatable sequences across
sessions.
norm = tf.random_normal([2, 3], seed=1234)
sess = tf.Session()
print(sess.run(norm))
print(sess.run(norm))
sess = tf.Session()
print(sess.run(norm))
print(sess.run(norm))
```

Another common use of random values is the initialization of variables. Also see the [Variables How To](#).

```
# Use random uniform values in [0, 1) as the initializer for a
variable of shape
# [2, 3]. The default type is float32.
var = tf.Variable(tf.random_uniform([2, 3]), name="var")
init = tf.global_variables_initializer()

sess = tf.Session()
sess.run(init)
print(sess.run(var))
```

- `tf.random_normal`
- `tf.truncated_normal`

- `tf.random_uniform`
- `tf.random_shuffle`
- `tf.random_crop`
- `tf.multinomial`
- `tf.random_gamma`
- `tf.set_random_seed`

Constantes, secuencias y valores aleatorios

Nota: Las funciones que toman `Tensor` argumentos también pueden tomar cualquier cosa aceptada por [`tf.convert_to_tensor`](#).

Tensores de valor constante

TensorFlow proporciona varias operaciones que puede usar para generar constantes.

- `tf.zeros`
- `tf.zeros_like`
- `tf.ones`
- `tf.ones_like`
- `tf.fill`
- `tf.constant`

Secuencias

- `tf.linspace`
- `tf.range`

Tensores aleatorios

TensorFlow tiene varias operaciones que crean tensores aleatorios con diferentes distribuciones. Las operaciones al azar son estables y crean nuevos valores aleatorios cada vez que se evalúan.

El `seed` argumento de palabra clave en estas funciones actúa en conjunción con la semilla aleatoria de nivel de gráfico. Cambiar la semilla de nivel de gráfico utilizando `tf.set_random_seed` la semilla de nivel operativo cambiará la semilla subyacente de estas operaciones. Al no establecer una semilla de nivel de gráfico ni de nivel operativo, se obtiene una semilla aleatoria para todas las operaciones. Consulte los `tf.set_random_seed` detalles sobre la interacción entre las semillas aleatorias de nivel de operación y de nivel de gráfico.

Ejemplos:

```
# Create a tensor of shape [2, 3] consisting of random normal values,
with mean
# -1 and standard deviation 4.
norm = tf.random_normal([2, 3], mean=-1, stddev=4)

# Shuffle the first dimension of a tensor
c = tf.constant([[1, 2], [3, 4], [5, 6]])
shuff = tf.random_shuffle(c)

# Each time we run these ops, different results are generated
sess = tf.Session()
print(sess.run(norm))
print(sess.run(norm))

# Set an op-level seed to generate repeatable sequences across
sessions.
norm = tf.random_normal([2, 3], seed=1234)
sess = tf.Session()
print(sess.run(norm))
print(sess.run(norm))
sess = tf.Session()
print(sess.run(norm))
print(sess.run(norm))
```

Otro uso común de los valores aleatorios es la inicialización de las variables. También vea las [variables Cómo hacer](#) .

```
# Use random uniform values in [0, 1) as the initializer for a
variable of shape
# [2, 3]. The default type is float32.
var = tf.Variable(tf.random_uniform([2, 3]), name="var")
init = tf.global_variables_initializer()

sess = tf.Session()
sess.run(init)
print(sess.run(var))
```

- `tf.random_normal`
- `tf.truncated_normal`
- `tf.random_uniform`
- `tf.random_shuffle`
- `tf.random_crop`
- `tf.multinomial`
- `tf.random_gamma`
- `tf.set_random_seed`