Creating Tensors and Operations

Fast Campus
Start Deep Learning with Tensorflow

Creating Tensors

- Every tensor is an instance of the Tensor class.
- A tensor may contain numbers, strings, or Boolean values. Every element of a tensor must have the same type.
- Tensors can be created, transformed, and operated upon using functions of the tf package.
- Each element in the Tensor has the same data type, and the data type is always known. The shape might be only partially known.

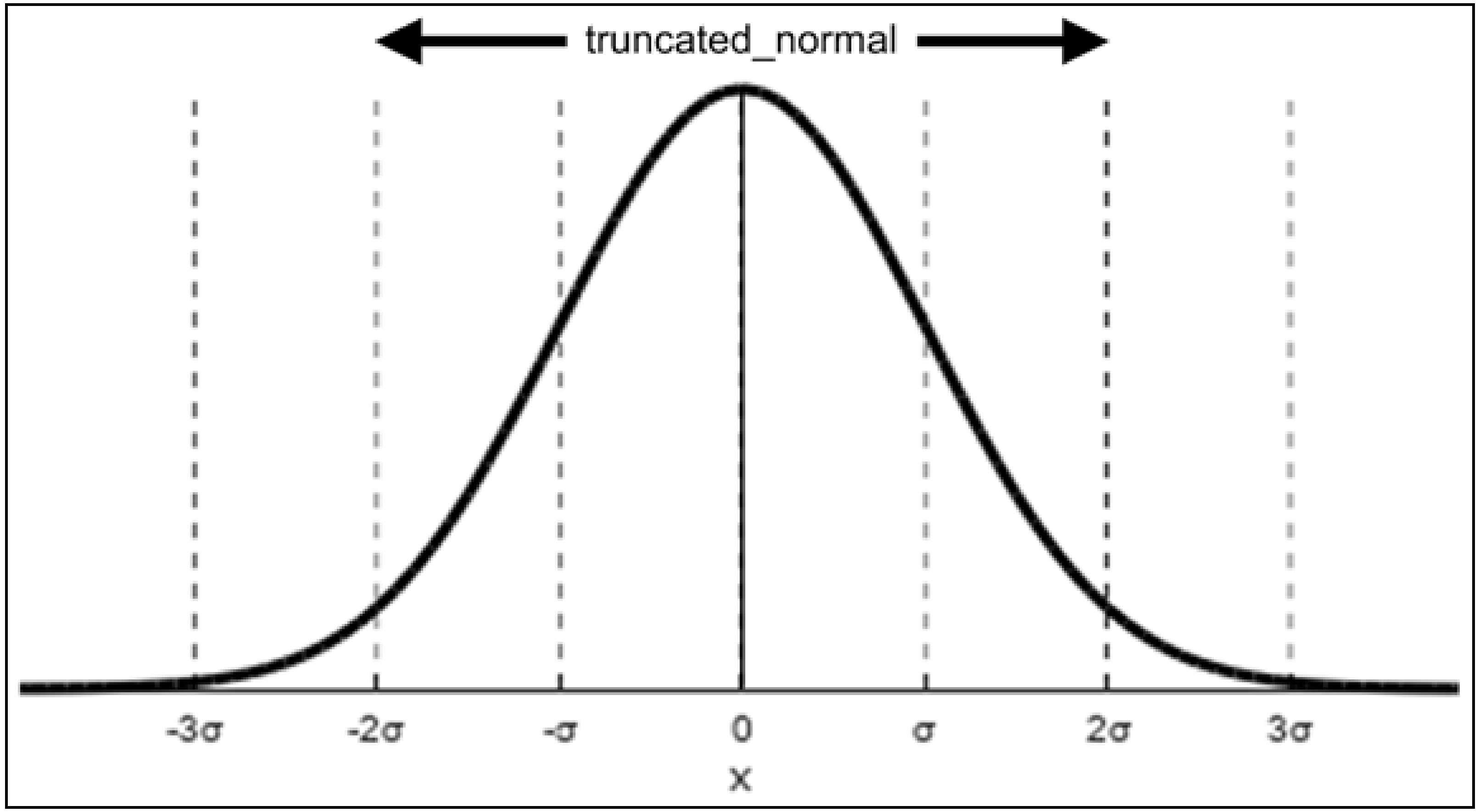
Creating Tensors with Known Values

Function	*Description*
`constant(value, dtype=None, shape = None, name = 'Const', verify_shape=False)`	Returns a tensor containing the given value
`zeros(shape, dtype=tf.float32, name = None)`	Returns a tensor filled with zeros
`ones(shape, dtype=tf.float32, name=None)`	Returns a tensor filled with ones
`fill(dims, value, name=None)`	Returns a tensor filled with the given value
`linspace(start, stop, num, name=None)`	Returns a tensor containing a linear range of values
`range(start, limit, delta=1, dtype=None, name='range')`	Returns a tensor containing a range of values
`range(limit, delta=1, dtype=None, name='range')`	Returns a tensor containing a range of values

Creating Tensors with Random Variables

Function	***Description***
`random_normal(shape, mean=0.0, stddev=1.0, dtype=tf.float32, seed=None, name=None)`	Creates a tensor with normally distributed values
`truncated_normal(shape, mean=0.0, stddev=1.0, dtype=tf.float32, seed=None, name=None)`	Creates a tensor with normally distributed values excluding those lying outside two standard deviations
`random_uniform(shape, minval=0, maxval=None, dtype=tf.float32, seed=None, name=None)`	Creates a tensor with uniformly distributed values between the minimum and maximum values
`random_shuffle(tensor, seed=None, name=None)`	Shuffles a tensor along its first dimension
`set_random_seed(seed)`	Set the seed value for all random number generation in the graph

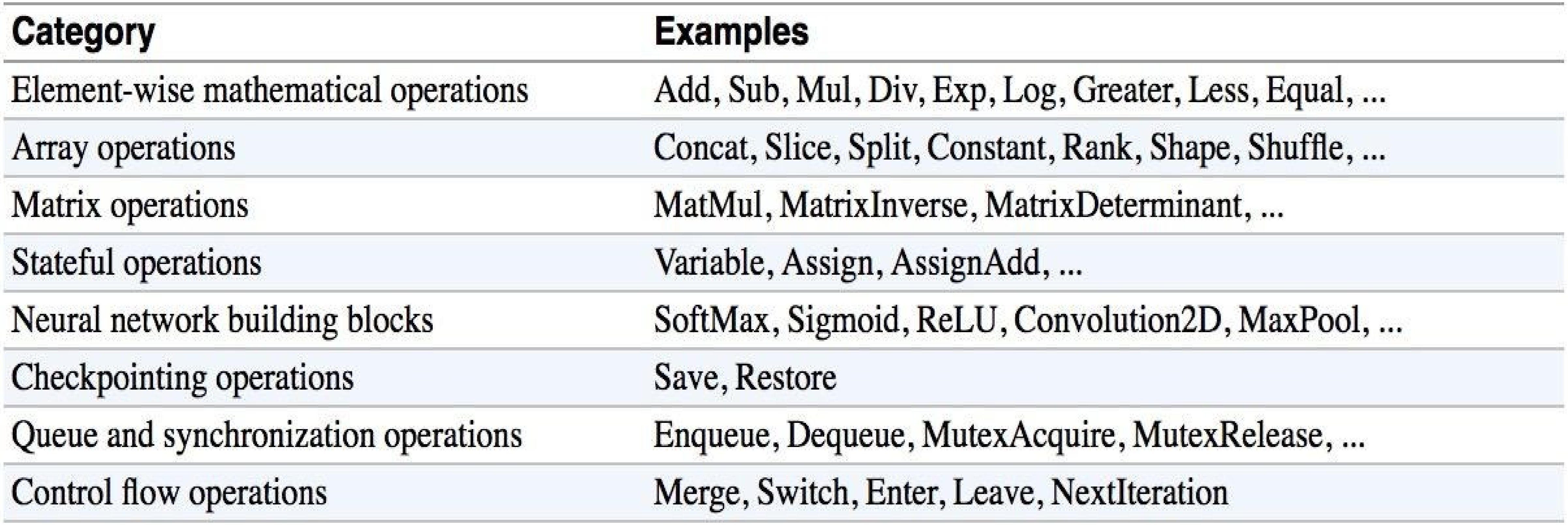
random_normal & truncated_normal



Functions for Transforming Tensors

Function	*Description*
`cast(tensor, dtype, name=None)`	Changes the tensor's data type to the given type
`reshape(tensor, shape, name=None)`	Returns a tensor with the same elements as the given tensor with the given shape
`squeeze(tensor, axis=None, name=None, squeeze_dims=None)`	Removes dimensions of size 1
`reverse(tensor, axis, name=None)`	Reverses given dimensions of the tensor
`slice(tensor, begin, size, name=None)`	Extracts a portion of a tensor
`stack(tensors, axis=0, name='stack')`	Combines a list of tensors into a tensor of greater rank

Operations



Category	Examples
Element-wise mathematical operations	Add, Sub, Mul, Div, Exp, Log, Greater, Less, Equal, ...
Array operations	Concat, Slice, Split, Constant, Rank, Shape, Shuffle, ...
Matrix operations	MatMul, MatrixInverse, MatrixDeterminant, ...
Stateful operations	Variable, Assign, AssignAdd, ...
Neural network building blocks	SoftMax, Sigmoid, ReLU, Convolution2D, MaxPool, ...
Checkpointing operations	Save, Restore
Queue and synchronization operations	Enqueue, Dequeue, MutexAcquire, MutexRelease, ...
Control flow operations	Merge, Switch, Enter, Leave, NextIteration

Wizard of Div

```
a = tf.constant([2, 2], name='a')
b = tf.constant([[0, 1], [2, 3]], name='b')
with tf.Session() as sess:
      print(sess.run(tf.div(b, a)))              ⇒ [[0 0] [1 1]]
      print(sess.run(tf.divide(b, a)))           ⇒ [[0. 0.5] [1. 1.5]]
      print(sess.run(tf.truediv(b, a)))          ⇒ [[0. 0.5] [1. 1.5]]
      print(sess.run(tf.floordiv(b, a)))         ⇒ [[0 0] [1 1]]
      print(sess.run(tf.realdiv(b, a)))          ⇒ # Error: only works for real
values
```