Lesson 18: More on Tensor

Variable Tensor

A variable tensor is created using ***tf.Variable()*** function.

Syntax: *tf.Variable(initial_value=None, trainable=None, validate_shape=True, caching_device=None, name=None, variable_def=None, dtype=None, import_scope=None, constraint=None, synchronization=tf.VariableSynchronization.AUTO, aggregation=tf.compat.v1.VariableAggregation.NONE, shape=None)*

- ***initial_value****: by default None. The initial value for the Variable is a Tensor, or a Python object convertible to a Tensor.*
- ***trainable****: by default None. If True, GradientTapes will keep an eye on this variable's usage.*
- ***validate_shape****: by default True. Allows the variable to be initialised with an unknown shape value if False. The shape of initial value must be known if True, which is the default.*
- ***name****:by default None. The variable's optional name. Defaults to 'Variable' and is automatically uniquified.*
- ***variable_def:*** *by default None.*
- ***dtype:*** *by default None. If set, initial_value will be converted to the given type. If None, either the datatype will be kept (if initial_value is a Tensor), or convert_to_tensor will decide.*
- ***shape:*** *by default None. if None the shape of initial_value will be used. if any shape is specified, the variable will be assigned with that particular shape.*

Few examples of Variable Tensor

```
import tensorflow as tf

x = tf.Variable( [1, 2, 3, 4] )

x = tf.Variable( [1.2, 4.4, 5, 6] )

x = tf.Variable( ['a', 'b', 'c', 'd'] )

x = tf.Variable( [True, False] )

x = tf.Variable( [3 + 4j] )
```

```
import tensorflow as tf

x = tf.Variable([1,2,3,4])
print(x)
```

```
<tf.Variable 'Variable:0' shape=(4,) dtype=int32, numpy=array([1, 2, 3, 4])>
```

```
x = tf.Variable([[1,2,3,4],[5,6,7,8]])
print(x)
```

```
<tf.Variable 'Variable:0' shape=(2, 4) dtype=int32, numpy= array([[1, 2, 3, 4],
                                                                  [5, 6, 7, 8]])>
```

Find the attributes of Tensor

```
x = tf.Variable([1,2,3,4])
print(x.name)

print(x.shape)

print(x.dtype)

print(x.numpy())
```

```
Variable:0
(4,)
<dtype: 'int32'>
[1 2 3 4]
```

```
x = tf.Variable([[1,2,3,4],[5,6,7,8]])
print(x.name)

print(x.shape)

print(x.dtype)

print(x.numpy())
```

```
Variable:0
(2, 4)
<dtype: 'int32'>
[[1 2 3 4]
 [5 6 7 8]]
```

Find the attributes of Tensor

```
x = tf.Variable([1,2,3,4])
print(x.name)

print(x.shape)

print(x.dtype)

print(x.numpy())
```

```
Variable:0
(4,)
<dtype: 'int32'>
[1 2 3 4]
```

```
x = tf.constant([1,2,3,4])
#print(x.name)      #possible, when eager execution is disabled

print(x.shape)

print(x.dtype)

print(x.numpy())
```

```
(4,)
<dtype: 'int32'>
[1 2 3 4]
```

Constant tensor can be converted to Variable tensor

```
x_con = tf.constant([1,2,3,4])

x_var = tf.Variable(t_con)
print(x_var)
```

```
<tf.Variable 'Variable:0' shape=(4,) dtype=int32, numpy=array([1, 2, 3, 4])>
```

Variable tensor can be converted to Constant tensor

```
x_var = tf.Variable([1,2,3,4])
x_con = tf.constant(x_var)
print(x_con)
```

```
tf.Tensor([1 2 3 4], shape=(4,), dtype=int32)
```

Include data type as parameter

```
x = tf.constant([1,2,3,4], dtype=tf.float32)
print(x)
```

```
tf.Tensor([1. 2. 3. 4.], shape=(4,), dtype=float32)
```