

tf-function

August 21, 2019

```
[1]: import tensorflow as tf
```

1 Simple usage

```
[2]: # define python function
def f(x):
    return 5.0 * x

print(f(5))
```

25.0

```
[3]: g = tf.function(f)

print(g(5))
```

tf.Tensor(25.0, shape=(), dtype=float32)

Note that `tf.function` doesn't automatically convert inputs to tensors. "... a new graph is generated for each distinct python numerical value, for example calling `g(2)` and `g(3)` will generate two new graphs (while only one is generated if you call `g(tf.constant(2))` and `g(tf.constant(3))`)."

2 As decorator

- automatically include functions within annotated function into graph
- automatically convert python flow control
- use `input_signature` to convert inputs to specified `TensorSpecs`

```
[4]: @tf.function(input_signature=[tf.TensorSpec(shape=[], dtype=tf.float32)])
def g2(x):
    if x < 3:
        return 5.0 * x
    else:
        # don't have to annotate f(x) separately
        return x * f(x)
```

```
print(g2(2))  
print(g2(5))
```

```
tf.Tensor(10.0, shape=(), dtype=float32)  
tf.Tensor(125.0, shape=(), dtype=float32)
```