tensorflow / tensorflow

TensorFlow: Open sou

Clone this repo:

```
git clone https://tensor
```

Branches

master

db0b

10.3 TensorFlow的自动求导机制

中国大学MOOC

tensorflow / tensorflow

TensorFlow: Open sou

Clone this repo:

```
git clone https://tensor
```

Branches

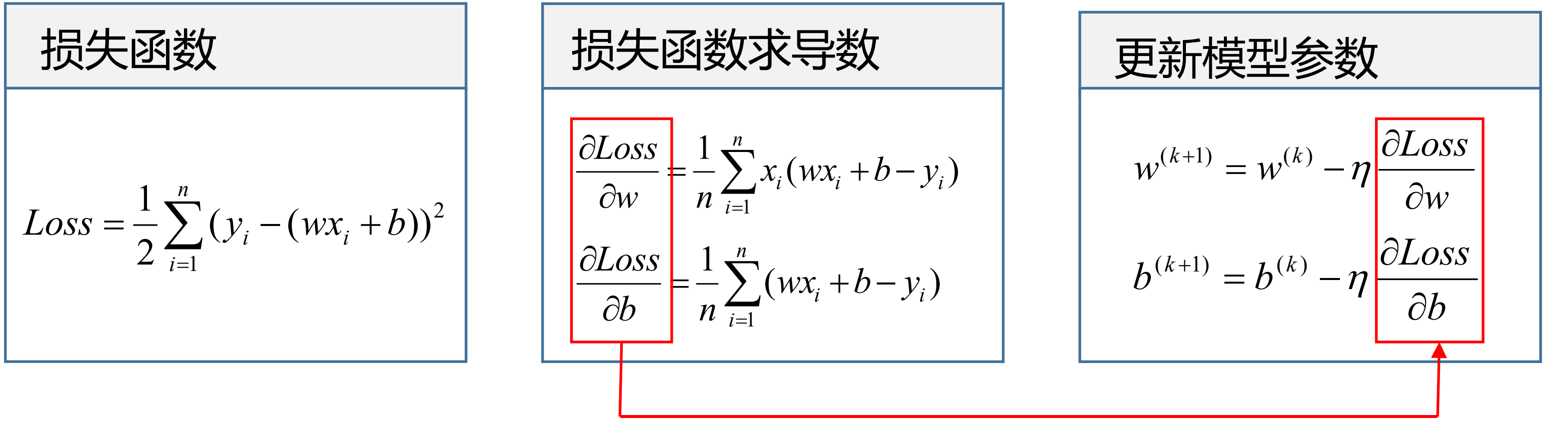
master

10.3.1 TensorFlow的可训练变量

中国大学MOOC

课程回顾

梯度下降法



损失函数

$$Loss = \frac{1}{2}\sum_{i=1}^{n}(y_i-(wx_i+b))^2$$

损失函数求导数

$$\frac{\partial Loss}{\partial w} = \frac{1}{n}\sum_{i=1}^{n}x_i(wx_i+b-y_i)$$

$$\frac{\partial Loss}{\partial b} = \frac{1}{n}\sum_{i=1}^{n}(wx_i+b-y_i)$$

更新模型参数

$$w^{(k+1)} = w^{(k)} - \eta\frac{\partial Loss}{\partial w}$$

$$b^{(k+1)} = b^{(k)} - \eta\frac{\partial Loss}{\partial b}$$

10.3.1 TensorFlow的自动求导机制——可训练变量

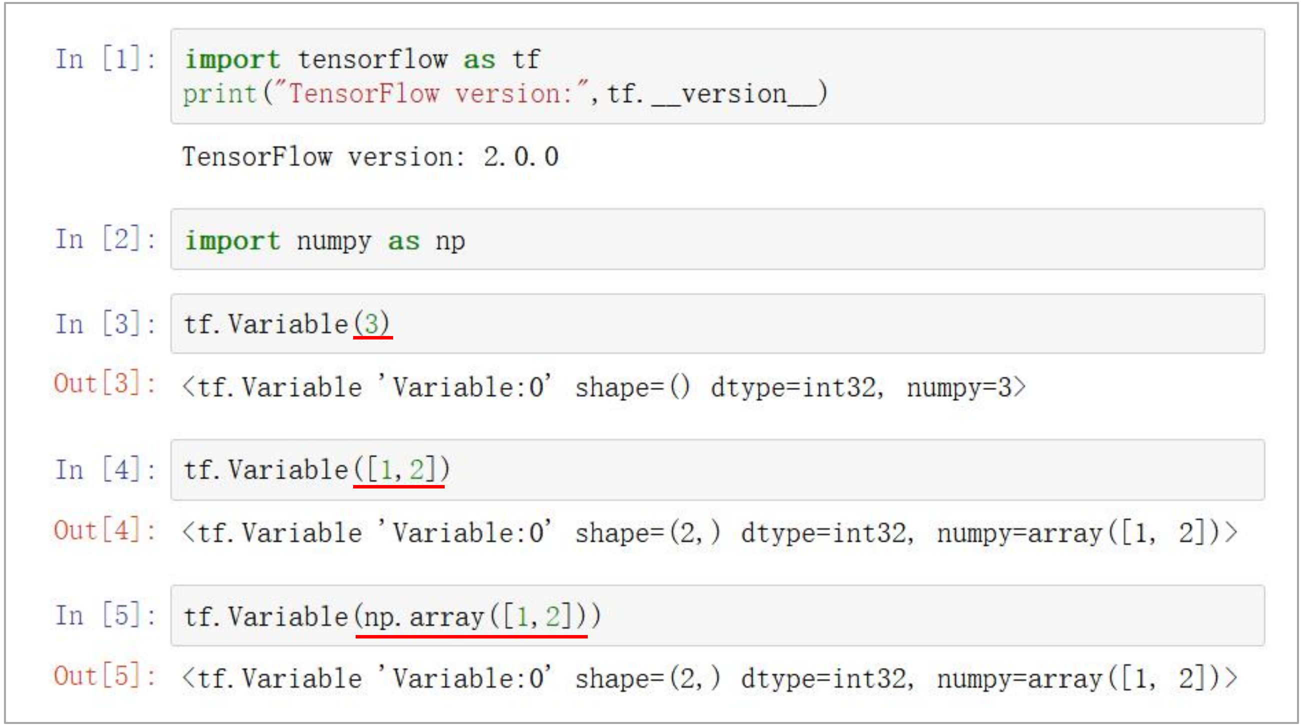
- TensorFlow的**自动求导**机制
- **Variable**对象
  - 对Tensor对象的进一步封装
  - 在模型训练过程中**自动记录梯度信息**，由算法**自动优化**
  - **可以被训练**的变量
  - 在机器学习中作为**模型参数**

```
tf.Variable(initial_value,dtype)
```

数字
Python列表
ndarray对象
Tensor对象

10.3.1 TensorFlow的自动求导机制——可训练变量

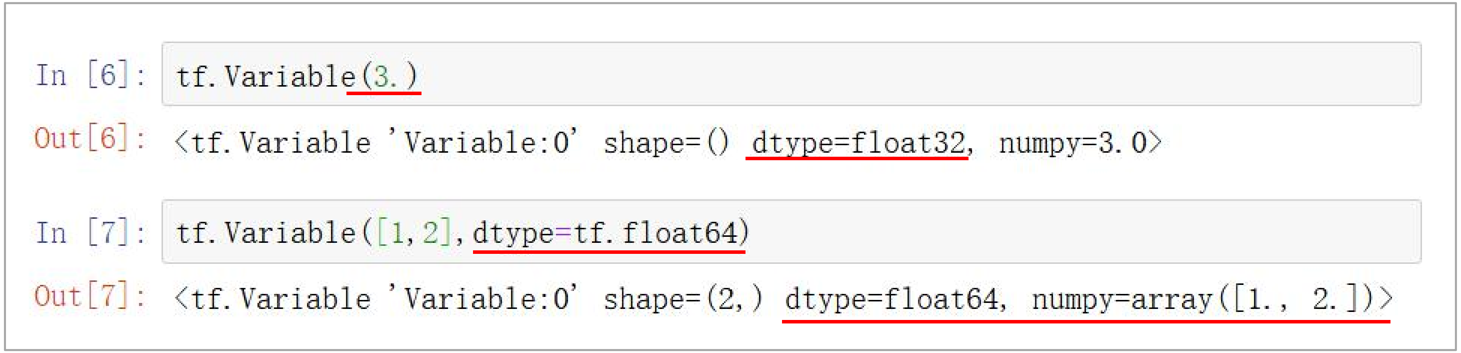
■ 创建**可训练变量**



```
In [1]:  import tensorflow as tf
         print("TensorFlow version:",tf.__version__)

         TensorFlow version: 2.0.0

In [2]:  import numpy as np

In [3]:  tf.Variable(3)

Out[3]:  <tf.Variable 'Variable:0' shape=() dtype=int32, numpy=3>

In [4]:  tf.Variable([1,2])

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

In [5]:  tf.Variable(np.array([1,2]))

Out[5]:  <tf.Variable 'Variable:0' shape=(2,) dtype=int32, numpy=array([1, 2])>
```

10.3.1 TensorFlow的自动求导机制——可训练变量

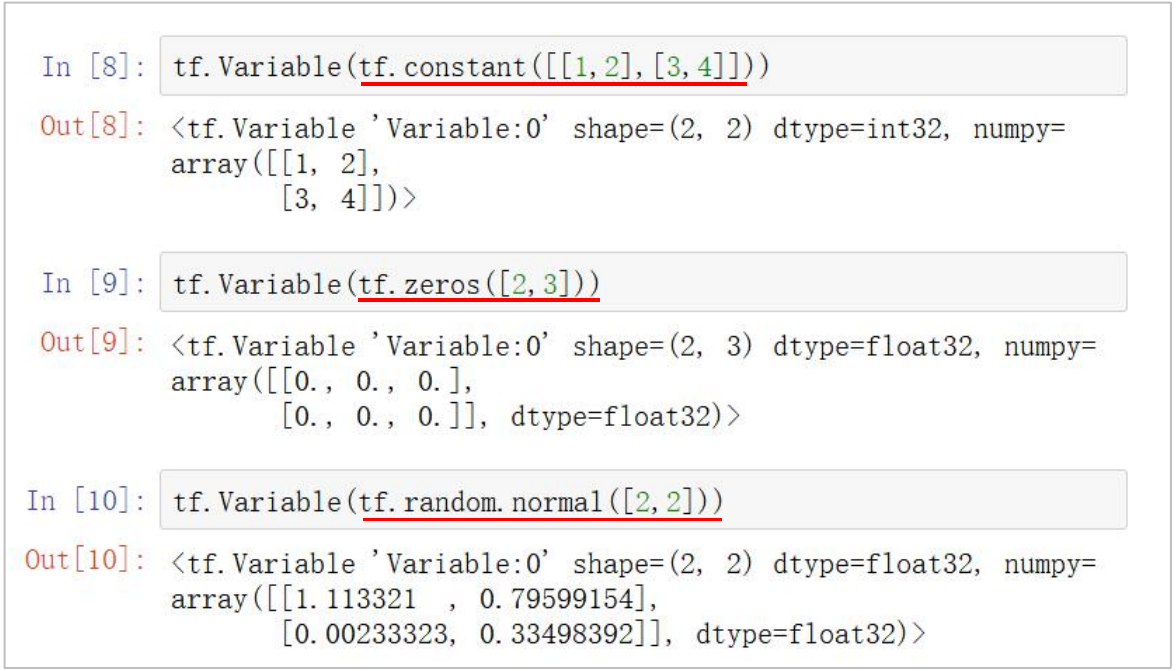
```
tf.Variable(initial_value,dtype)
```



```
In [6]:  tf.Variable(3.)

Out[6]:  <tf.Variable 'Variable:0' shape=() dtype=float32, numpy=3.0>

In [7]:  tf.Variable([1,2],dtype=tf.float64)

Out[7]:  <tf.Variable 'Variable:0' shape=(2,) dtype=float64, numpy=array([1., 2.])>
```

10.3.1 TensorFlow的自动求导机制——可训练变量

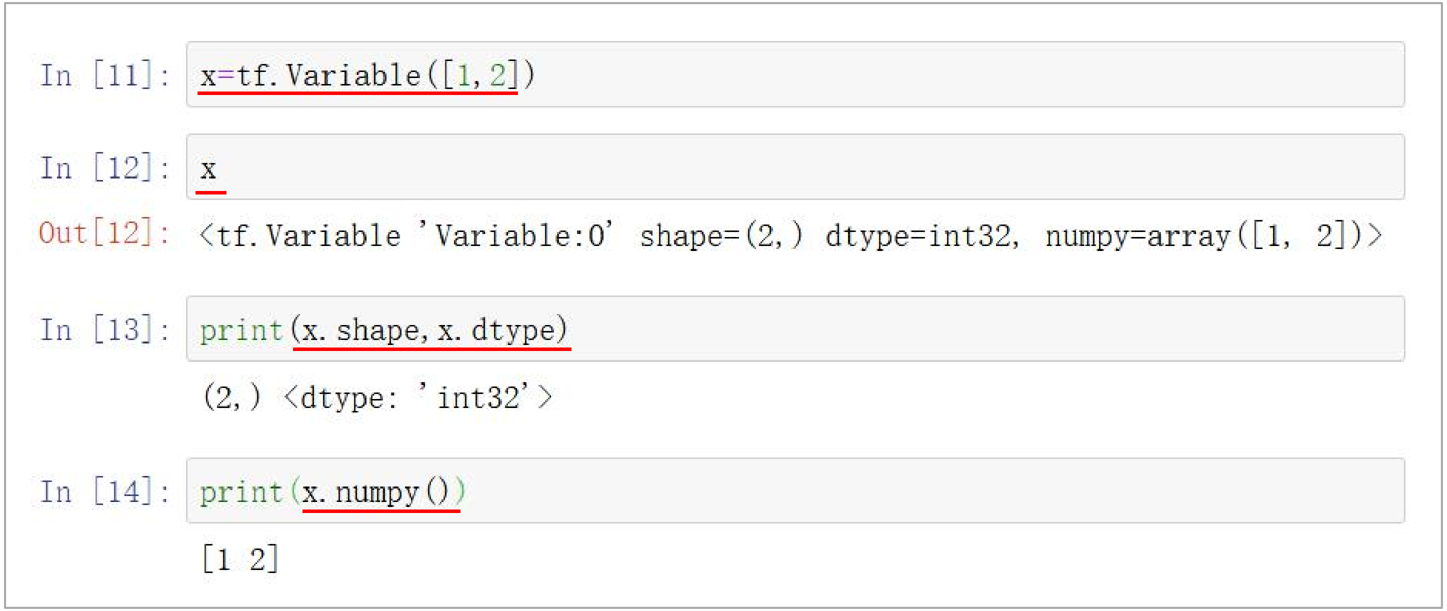
□ 将张量封装为可训练变量



```
In [8]:  tf.Variable(tf.constant([[1,2],[3,4]]))

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

In [9]:  tf.Variable(tf.zeros([2,3]))

Out[9]:  <tf.Variable 'Variable:0' shape=(2, 3) dtype=float32, numpy=
         array([[0., 0., 0.],
                [0., 0., 0.]], dtype=float32)>

In [10]: tf.Variable(tf.random.normal([2,2]))

Out[10]: <tf.Variable 'Variable:0' shape=(2, 2) dtype=float32, numpy=
         array([[1.113321  , 0.79599154],
                [0.00233323, 0.33498392]], dtype=float32)>
```

10.3.1 TensorFlow的自动求导机制——可训练变量

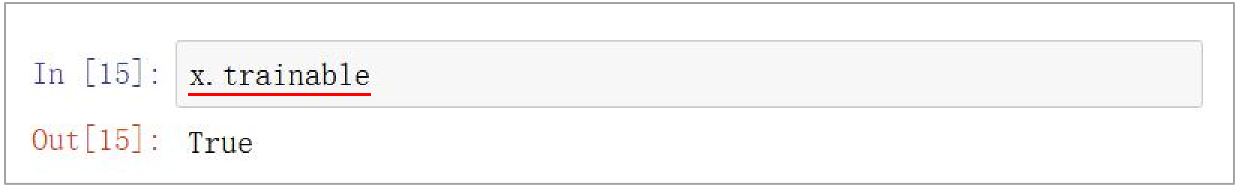
■ 使用**变量名**



```
In [11]: x=tf.Variable([1,2])

In [12]: x

Out[12]: <tf.Variable 'Variable:0' shape=(2,) dtype=int32, numpy=array([1, 2])>

In [13]: print(x.shape,x.dtype)

         (2,) <dtype: 'int32'>

In [14]: print(x.numpy())

         [1 2]
```

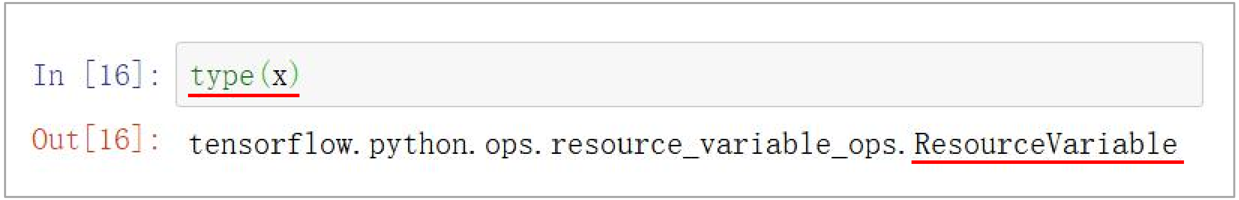
10.3.1 TensorFlow的自动求导机制——可训练变量

trainalbe属性



```
In [15]: x.trainable

Out[15]: True
```

ResourceVariable

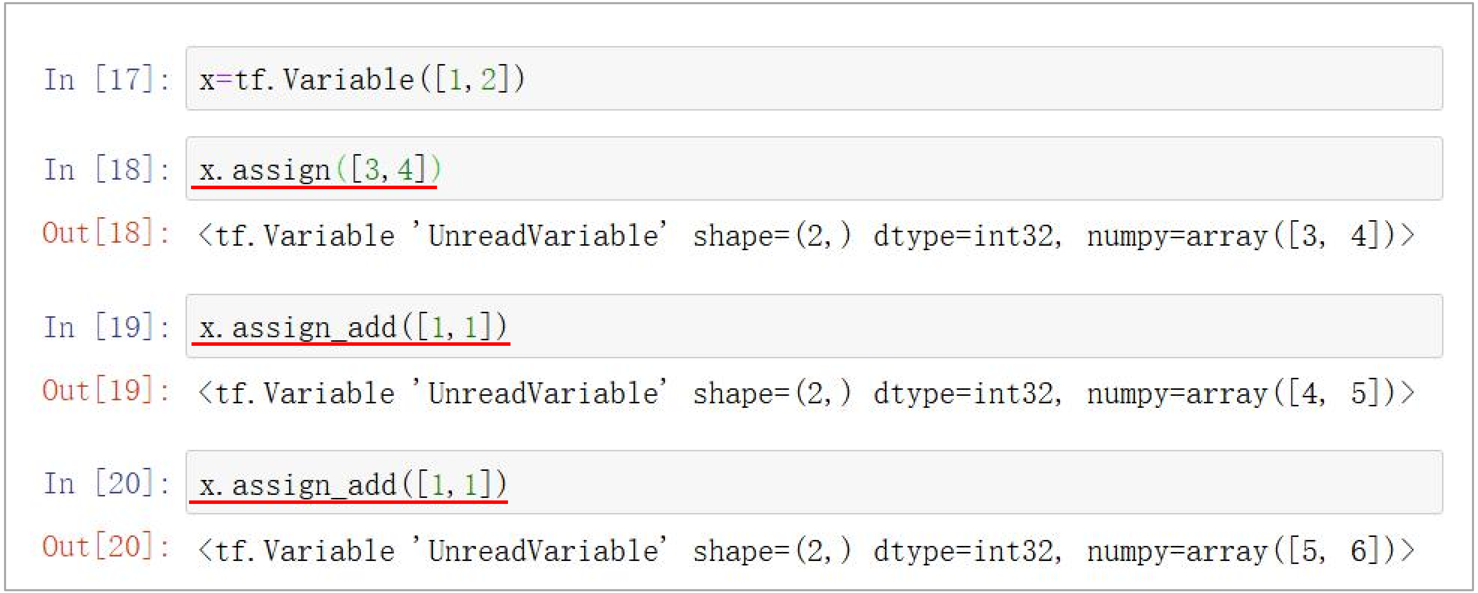


```
In [16]: type(x)

Out[16]: tensorflow.python.ops.resource_variable_ops.ResourceVariable
```

10.3.1 TensorFlow的自动求导机制——可训练变量

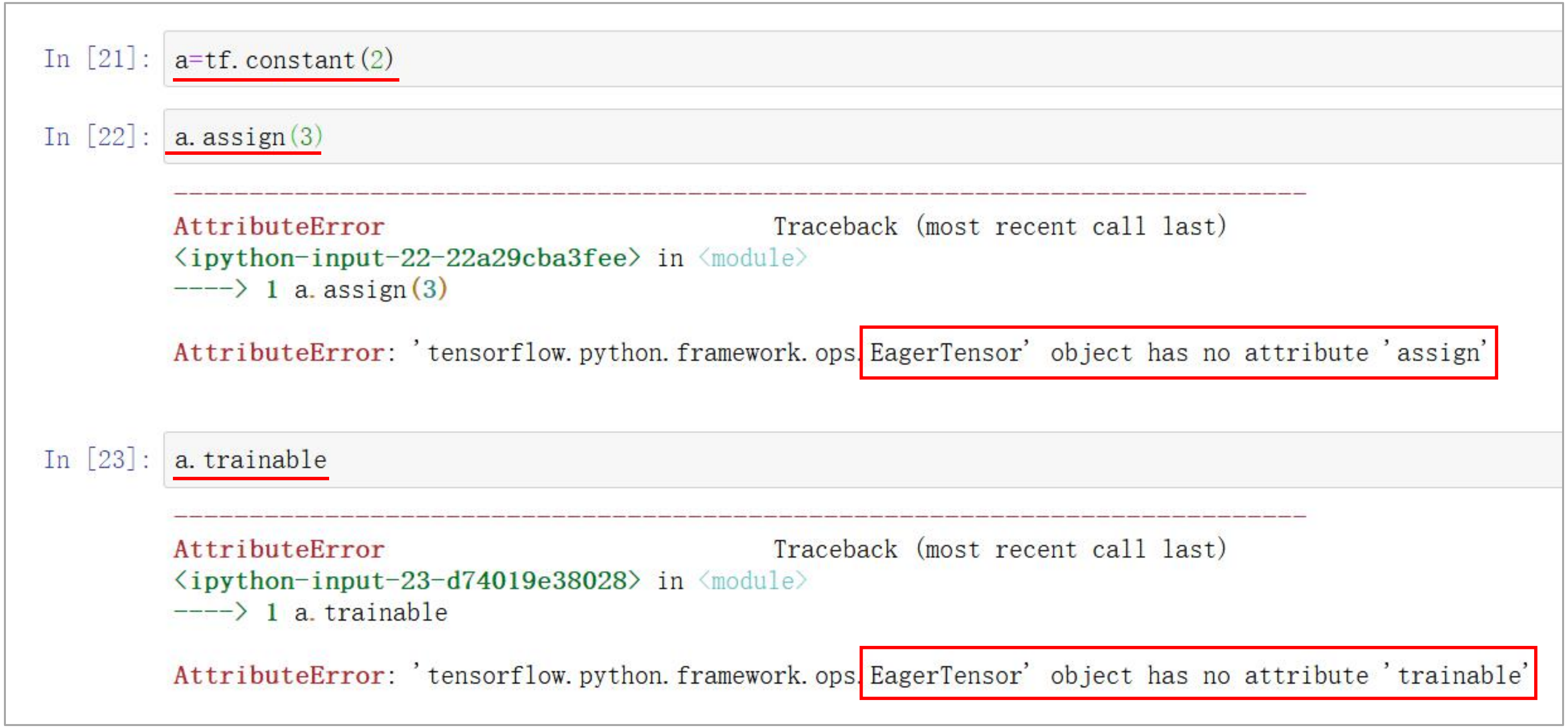
可训练变量**赋值**

对象名.assign() 对象名.assign_add() 对象名.assign_sub()



```
In [17]: x=tf.Variable([1,2])

In [18]: x.assign([3,4])
Out[18]: <tf.Variable 'UnreadVariable' shape=(2,) dtype=int32, numpy=array([3, 4])>

In [19]: x.assign_add([1,1])
Out[19]: <tf.Variable 'UnreadVariable' shape=(2,) dtype=int32, numpy=array([4, 5])>

In [20]: x.assign_add([1,1])
Out[20]: <tf.Variable 'UnreadVariable' shape=(2,) dtype=int32, numpy=array([5, 6])>
```

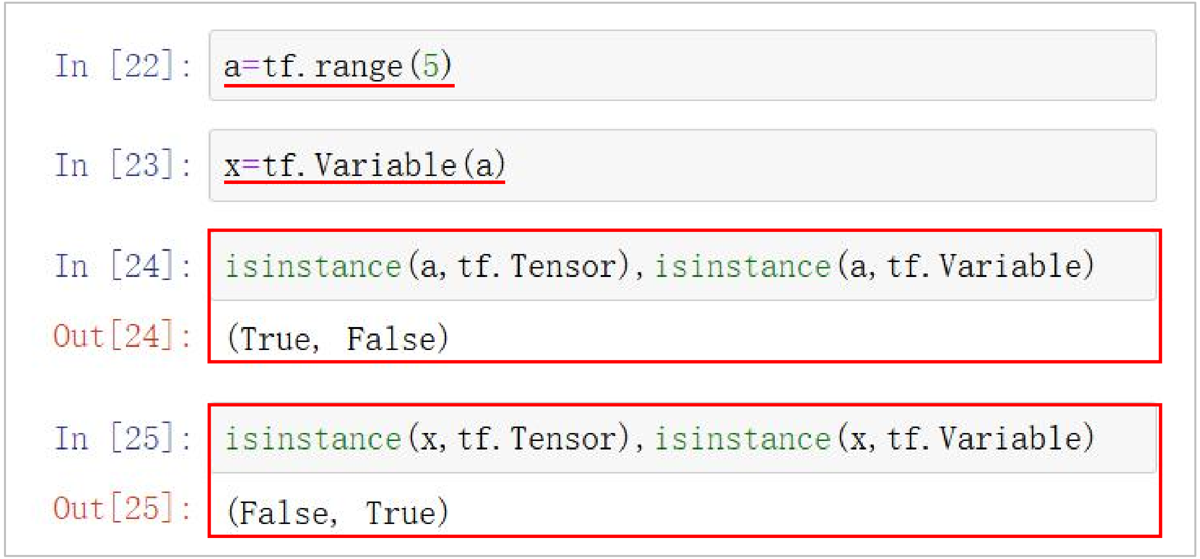
10.3.1 TensorFlow的自动求导机制——可训练变量



```
In [21]: a=tf.constant(2)

In [22]: a.assign(3)

---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-22-22a29cba3fee> in <module>
----> 1 a.assign(3)

AttributeError: 'tensorflow.python.framework.ops.EagerTensor' object has no attribute 'assign'

In [23]: a.trainable

---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-23-d74019e38028> in <module>
----> 1 a.trainable

AttributeError: 'tensorflow.python.framework.ops.EagerTensor' object has no attribute 'trainable'
```

10.3.1 TensorFlow的自动求导机制——可训练变量

■ isinstance() 方法



```
In [22]: a=tf.range(5)

In [23]: x=tf.Variable(a)

In [24]: isinstance(a,tf.Tensor),isinstance(a,tf.Variable)

Out[24]: (True, False)

In [25]: isinstance(x,tf.Tensor),isinstance(x,tf.Variable)

Out[25]: (False, True)
```

神经网络&深度学习 TENSORFLOW Google

10.3.2 TensorFlow的自动求导

中国大学MOOC

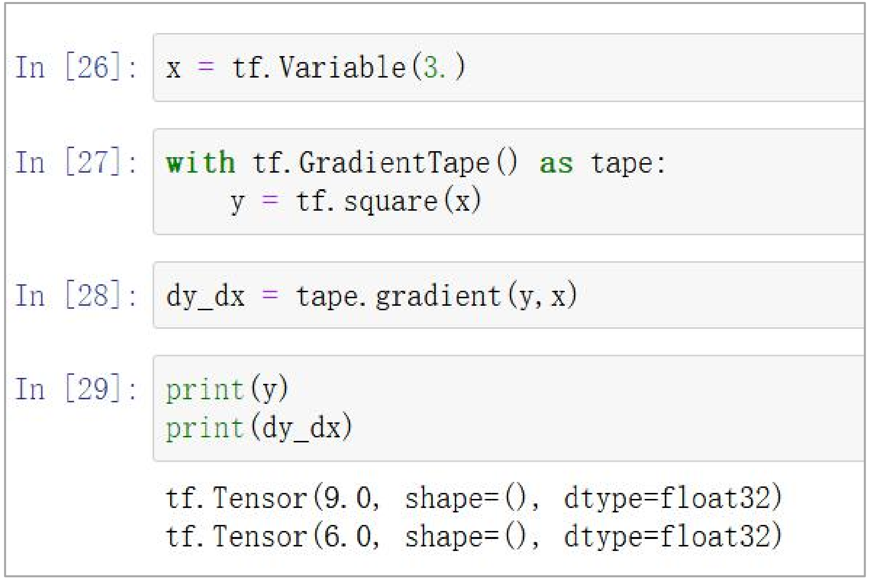
10.3.2 TensorFlow的自动求导

■ 自动求导——GradientTape

```
with GradientTape() as tape:
    函数表达式
grad=tape.gradient(函数,自变量)
```

$$y = x^2 \big|_{x=3}$$

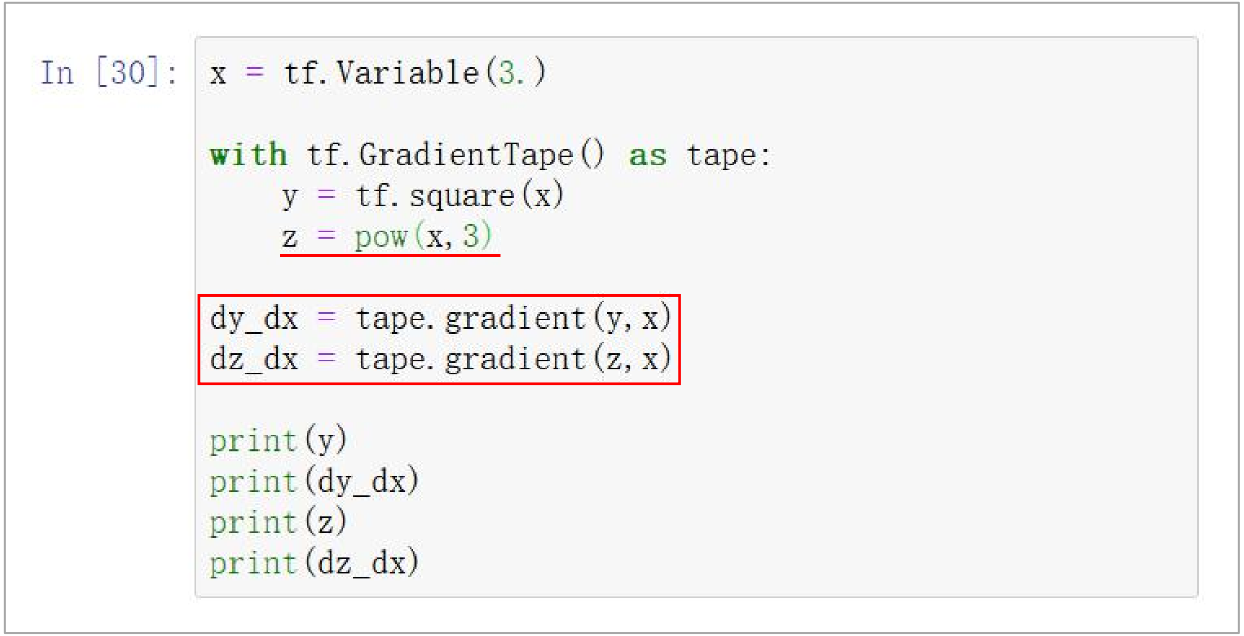
$$\boxed{\begin{aligned} y &= 3^2 = 9 \\ \frac{dy}{dx} &= 2x = 6 \end{aligned}}$$



```
In [26]: x = tf.Variable(3.)

In [27]: with tf.GradientTape() as tape:
             y = tf.square(x)

In [28]: dy_dx = tape.gradient(y, x)

In [29]: print(y)
         print(dy_dx)

         tf.Tensor(9.0, shape=(), dtype=float32)
         tf.Tensor(6.0, shape=(), dtype=float32)
```

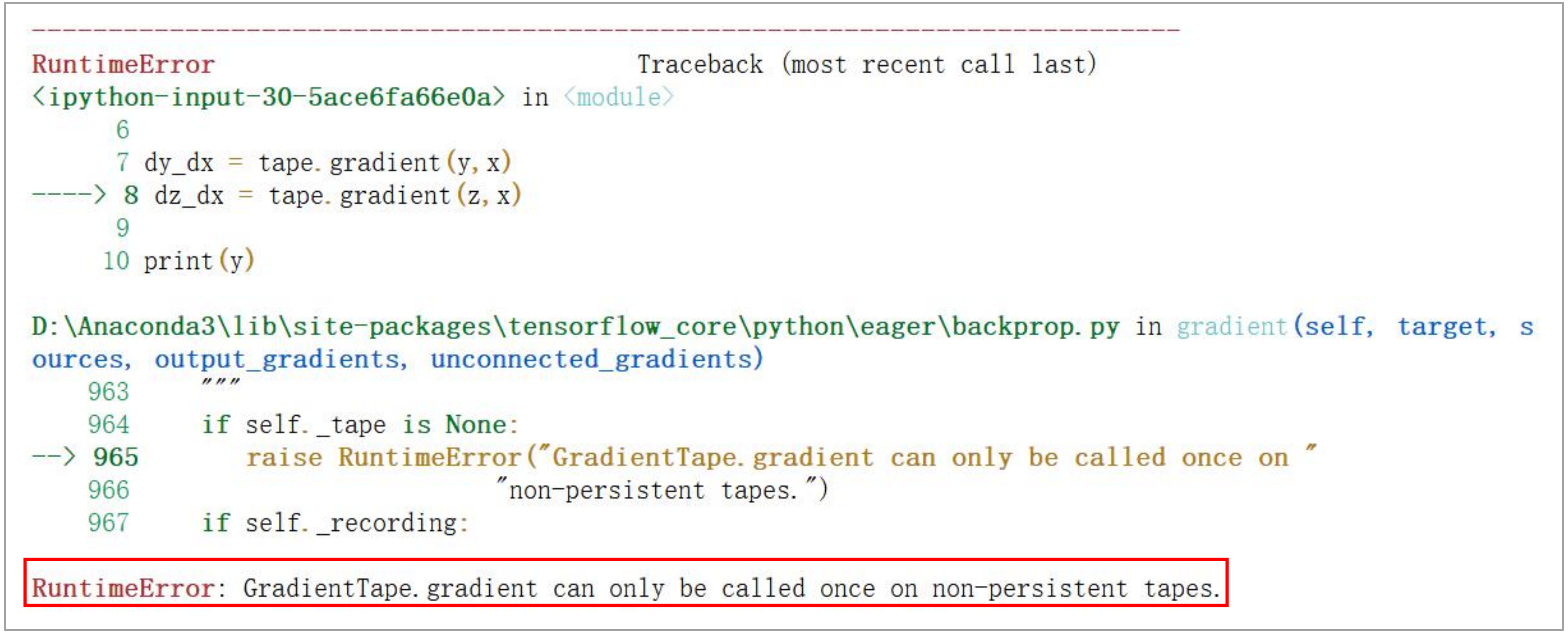
10.3.2 TensorFlow的自动求导

```
GradientTape(persistent, watch_accessed_variables)
```



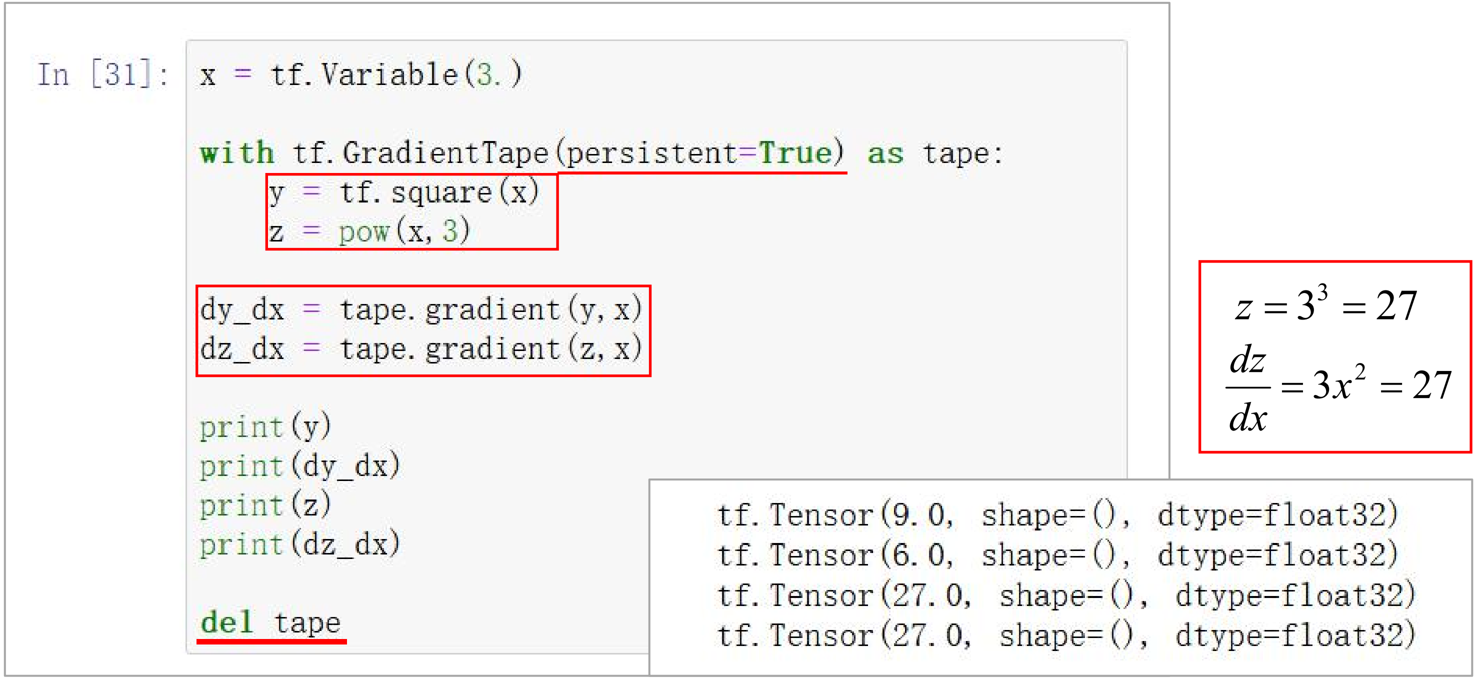
```
In [30]: x = tf.Variable(3.)

         with tf.GradientTape() as tape:
             y = tf.square(x)
             z = pow(x, 3)

         dy_dx = tape.gradient(y, x)
         dz_dx = tape.gradient(z, x)

         print(y)
         print(dy_dx)
         print(z)
         print(dz_dx)
```

10.3.2 TensorFlow的自动求导



```
---------------------------------------------------------------------------
RuntimeError                              Traceback (most recent call last)
<ipython-input-30-5ace6fa66e0a> in <module>
      6
      7 dy_dx = tape.gradient(y, x)
----> 8 dz_dx = tape.gradient(z, x)
      9
     10 print(y)

D:\Anaconda3\lib\site-packages\tensorflow_core\python\eager\backprop.py in gradient(self, target, s
ources, output_gradients, unconnected_gradients)
    963     """
    964     if self._tape is None:
--> 965       raise RuntimeError("GradientTape.gradient can only be called once on "
    966                          "non-persistent tapes.")
    967     if self._recording:

RuntimeError: GradientTape.gradient can only be called once on non-persistent tapes.
```

10.3.2 TensorFlow的自动求导



```
In [31]: x = tf.Variable(3.)

         with tf.GradientTape(persistent=True) as tape:
             y = tf.square(x)
             z = pow(x,3)

         dy_dx = tape.gradient(y,x)
         dz_dx = tape.gradient(z,x)

         print(y)
         print(dy_dx)
         print(z)
         print(dz_dx)

         del tape
```

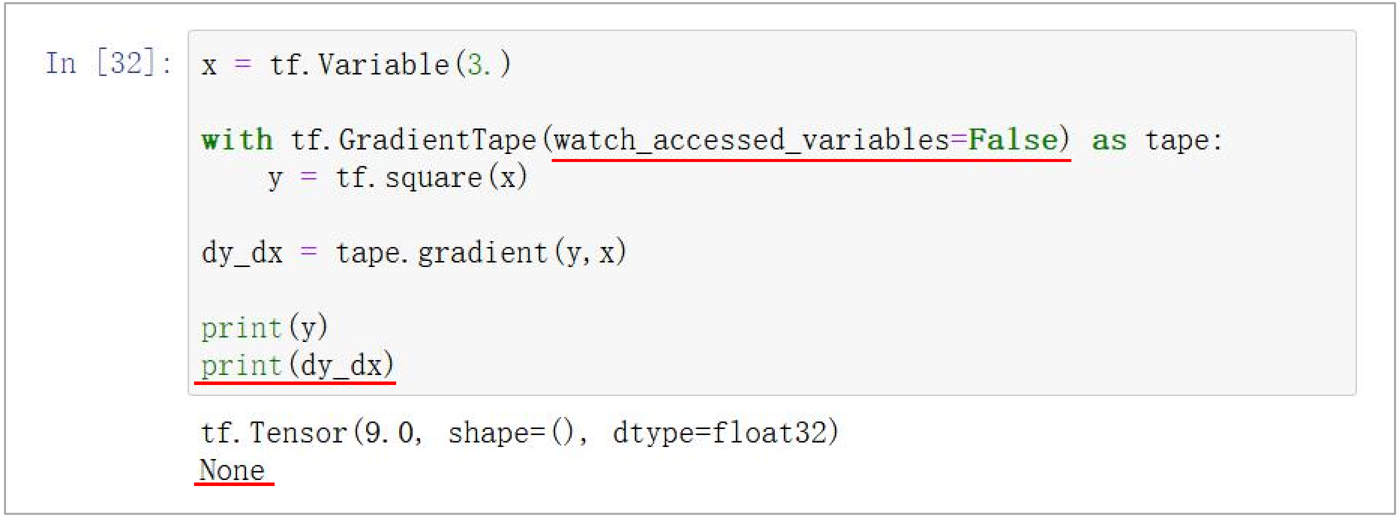
```
tf.Tensor(9.0, shape=(), dtype=float32)
tf.Tensor(6.0, shape=(), dtype=float32)
tf.Tensor(27.0, shape=(), dtype=float32)
tf.Tensor(27.0, shape=(), dtype=float32)
```

$$z = 3^3 = 27$$

$$\frac{dz}{dx} = 3x^2 = 27$$

10.3.2 TensorFlow的自动求导

```
GradientTape(persistent,watch_accessed_variables)
```

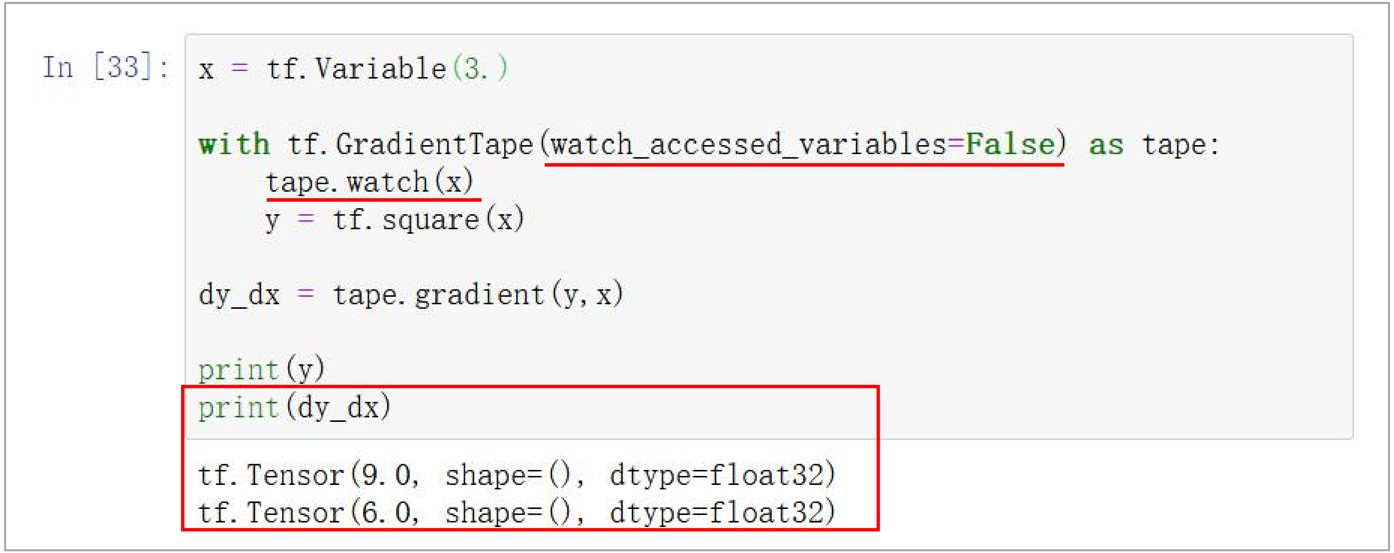


```
In [32]: x = tf.Variable(3.)

         with tf.GradientTape(watch_accessed_variables=False) as tape:
             y = tf.square(x)

         dy_dx = tape.gradient(y,x)

         print(y)
         print(dy_dx)
```

```
tf.Tensor(9.0, shape=(), dtype=float32)
None
```

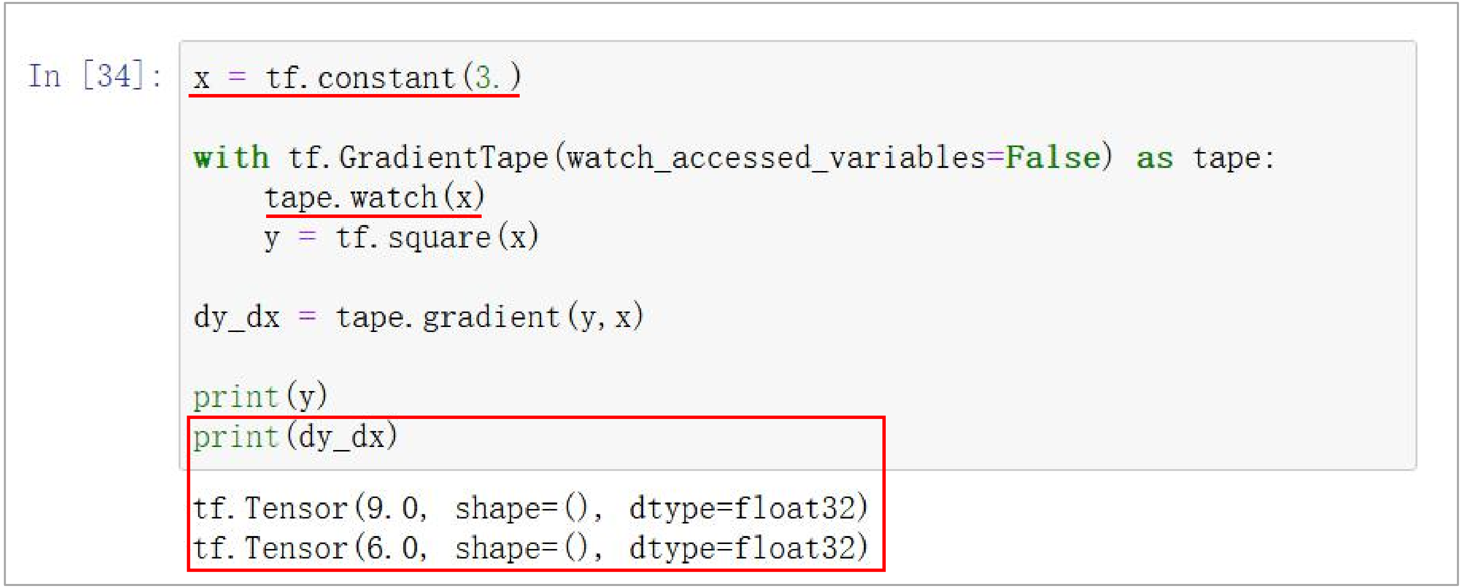
10.3.2 TensorFlow的自动求导

添加监视——watch()



```
In [33]: x = tf.Variable(3.)

         with tf.GradientTape(watch_accessed_variables=False) as tape:
             tape.watch(x)
             y = tf.square(x)

         dy_dx = tape.gradient(y, x)

         print(y)
         print(dy_dx)

         tf.Tensor(9.0, shape=(), dtype=float32)
         tf.Tensor(6.0, shape=(), dtype=float32)
```

监视非可训练变量



```
In [34]: x = tf.constant(3.)

         with tf.GradientTape(watch_accessed_variables=False) as tape:
             tape.watch(x)
             y = tf.square(x)

         dy_dx = tape.gradient(y,x)

         print(y)
         print(dy_dx)

         tf.Tensor(9.0, shape=(), dtype=float32)
         tf.Tensor(6.0, shape=(), dtype=float32)
```

10.3.2 TensorFlow的自动求导

多元函数求偏导数

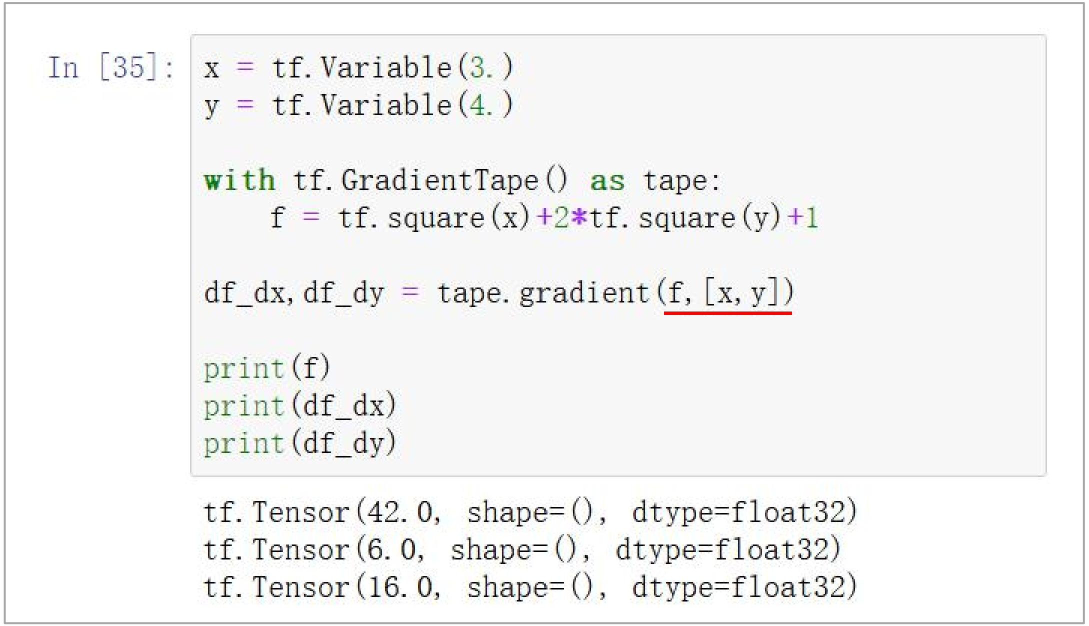
```
tape.gradient(函数, 自变量)
```

$$f(x,y)=x^2+2y^2+1$$

$$f(3,4)=42$$

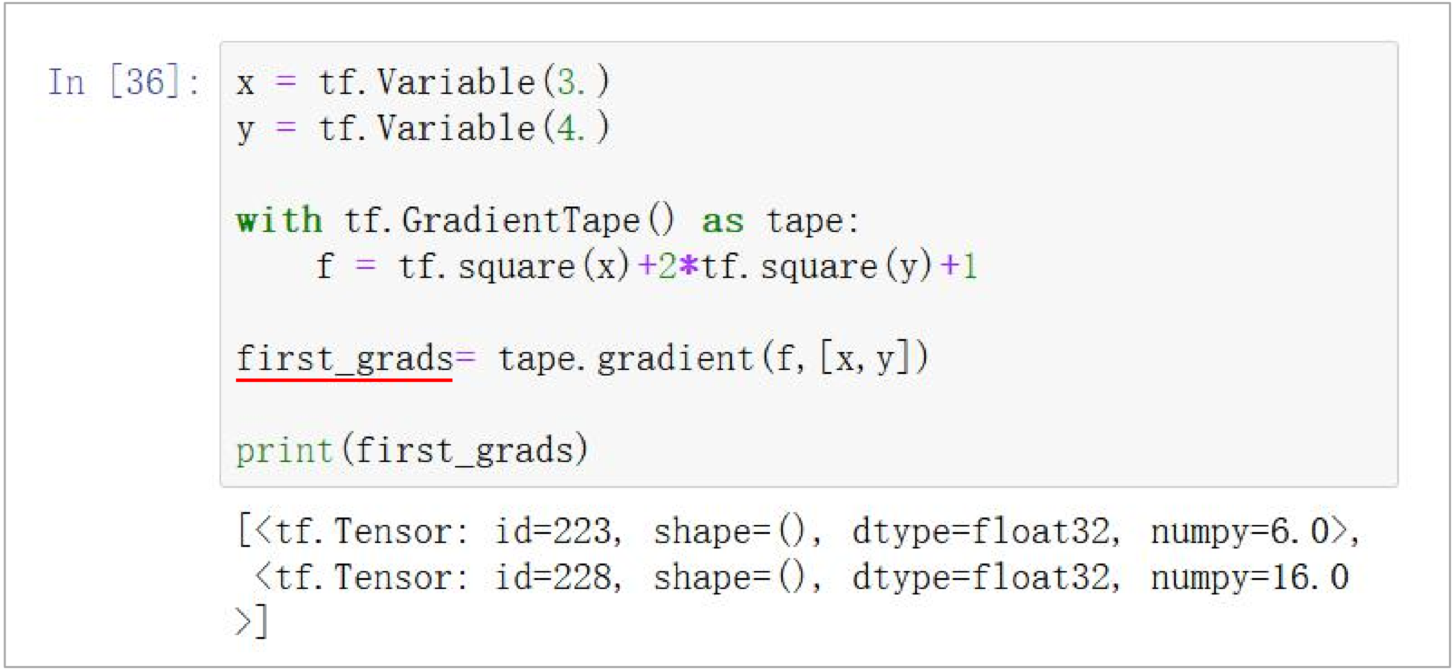
$$\frac{\partial f(x,y)}{\partial x}\Big|_{x=3}=2x=6$$

$$\frac{\partial f(x,y)}{\partial y}\Big|_{y=4}=4y=16$$



```
In [35]: x = tf.Variable(3.)
         y = tf.Variable(4.)

         with tf.GradientTape() as tape:
             f = tf.square(x)+2*tf.square(y)+1

         df_dx, df_dy = tape.gradient(f, [x, y])

         print(f)
         print(df_dx)
         print(df_dy)

         tf.Tensor(42.0, shape=(), dtype=float32)
         tf.Tensor(6.0, shape=(), dtype=float32)
         tf.Tensor(16.0, shape=(), dtype=float32)
```

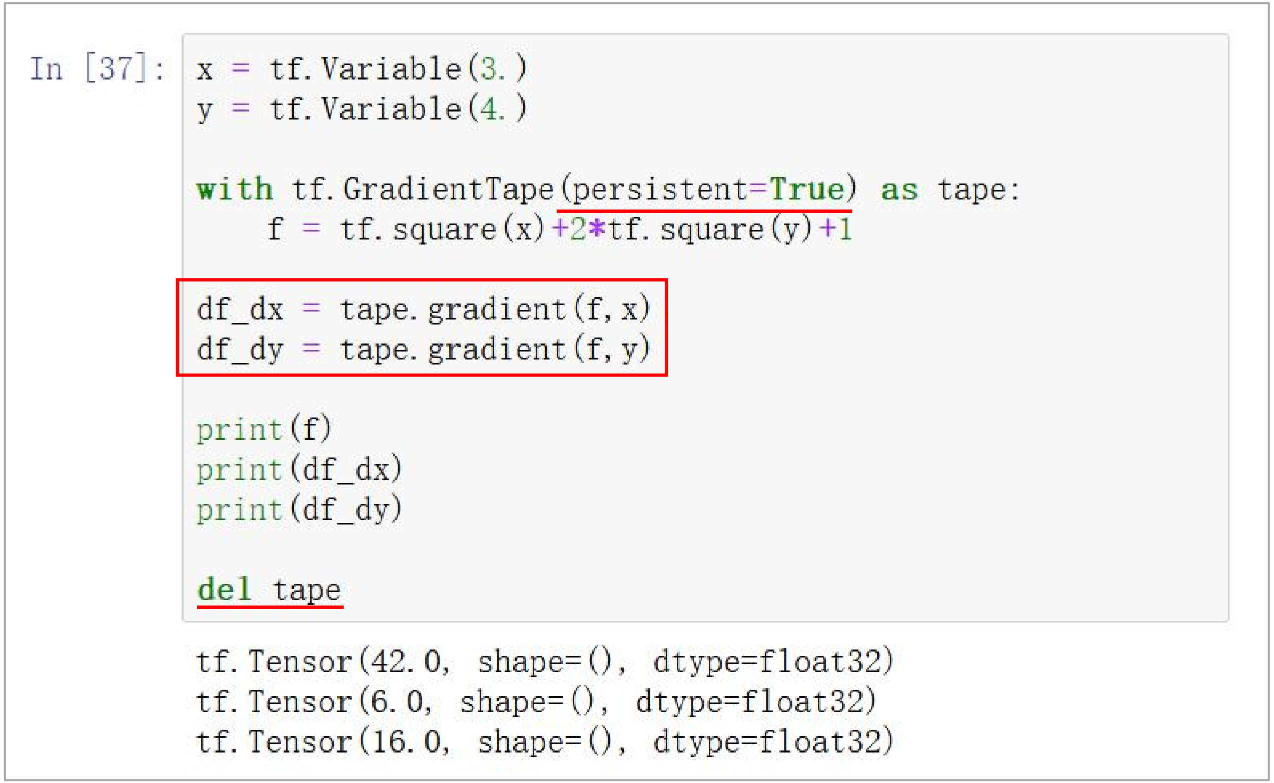
10.3.2 TensorFlow的自动求导



In [36]:
```
x = tf.Variable(3.)
y = tf.Variable(4.)

with tf.GradientTape() as tape:
    f = tf.square(x)+2*tf.square(y)+1

first_grads= tape.gradient(f,[x,y])

print(first_grads)
```

```
[<tf.Tensor: id=223, shape=(), dtype=float32, numpy=6.0>,
 <tf.Tensor: id=228, shape=(), dtype=float32, numpy=16.0
>]
```

10.3.2 TensorFlow的自动求导



```
In [37]: x = tf.Variable(3.)
         y = tf.Variable(4.)

         with tf.GradientTape(persistent=True) as tape:
             f = tf.square(x)+2*tf.square(y)+1

         df_dx = tape.gradient(f,x)
         df_dy = tape.gradient(f,y)

         print(f)
         print(df_dx)
         print(df_dy)

         del tape
```

```
tf.Tensor(42.0, shape=(), dtype=float32)
tf.Tensor(6.0, shape=(), dtype=float32)
tf.Tensor(16.0, shape=(), dtype=float32)
```

10.3.2 TensorFlow的自动求导

■ 求二阶导数

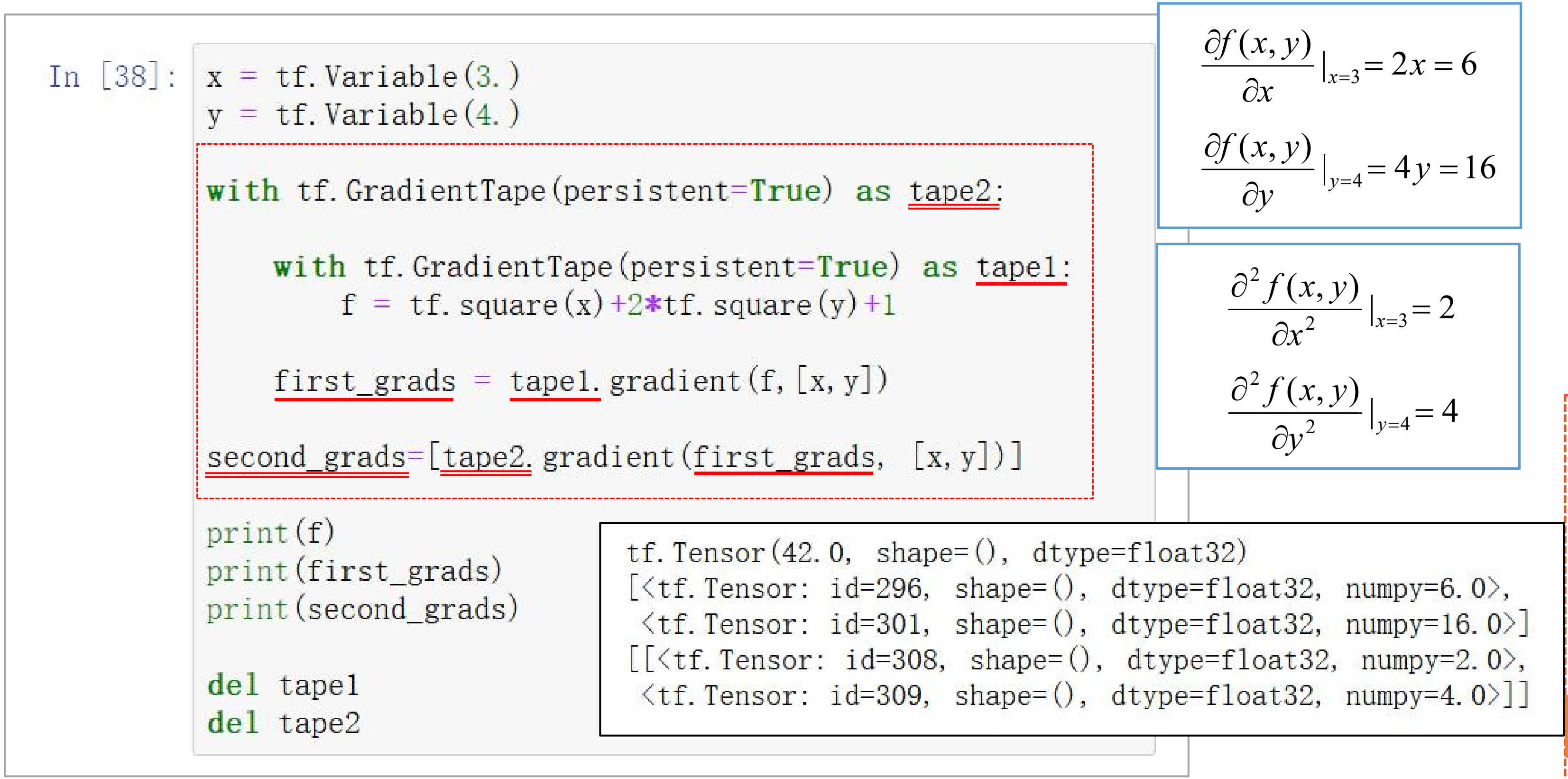
$$f(x,y)=x^2+2y^2+1$$

$$\frac{\partial f(x,y)}{\partial x}\Big|_{x=3}=2x=6$$

$$\frac{\partial f(x,y)}{\partial y}\Big|_{y=4}=4y=16$$

$$\frac{\partial^2 f(x,y)}{\partial x^2}\Big|_{x=3}=2$$

$$\frac{\partial^2 f(x,y)}{\partial y^2}\Big|_{y=4}=4$$

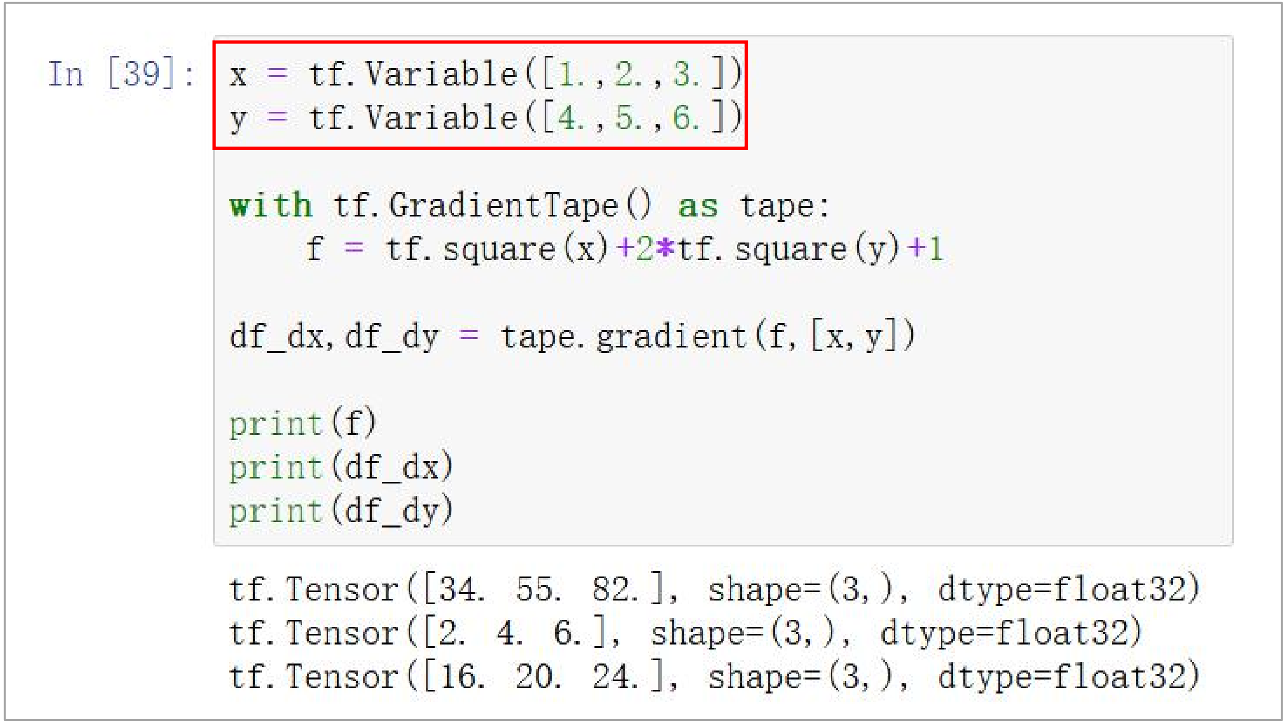


In [38]:
```
x = tf.Variable(3.)
y = tf.Variable(4.)

with tf.GradientTape(persistent=True) as tape2:

    with tf.GradientTape(persistent=True) as tape1:
        f = tf.square(x)+2*tf.square(y)+1

    first_grads = tape1.gradient(f, [x, y])

second_grads=[tape2.gradient(first_grads, [x, y])]

print(f)
print(first_grads)
print(second_grads)

del tape1
del tape2
```

```
tf.Tensor(42.0, shape=(), dtype=float32)
[<tf.Tensor: id=296, shape=(), dtype=float32, numpy=6.0>,
 <tf.Tensor: id=301, shape=(), dtype=float32, numpy=16.0>]
[[<tf.Tensor: id=308, shape=(), dtype=float32, numpy=2.0>,
 <tf.Tensor: id=309, shape=(), dtype=float32, numpy=4.0>]]
```

10.3.2 TensorFlow的自动求导

对向量求偏导



```
In [39]: x = tf.Variable([1.,2.,3.])
         y = tf.Variable([4.,5.,6.])

         with tf.GradientTape() as tape:
             f = tf.square(x)+2*tf.square(y)+1

         df_dx,df_dy = tape.gradient(f,[x,y])

         print(f)
         print(df_dx)
         print(df_dy)

         tf.Tensor([34. 55. 82.], shape=(3,), dtype=float32)
         tf.Tensor([2. 4. 6.], shape=(3,), dtype=float32)
         tf.Tensor([16. 20. 24.], shape=(3,), dtype=float32)
```