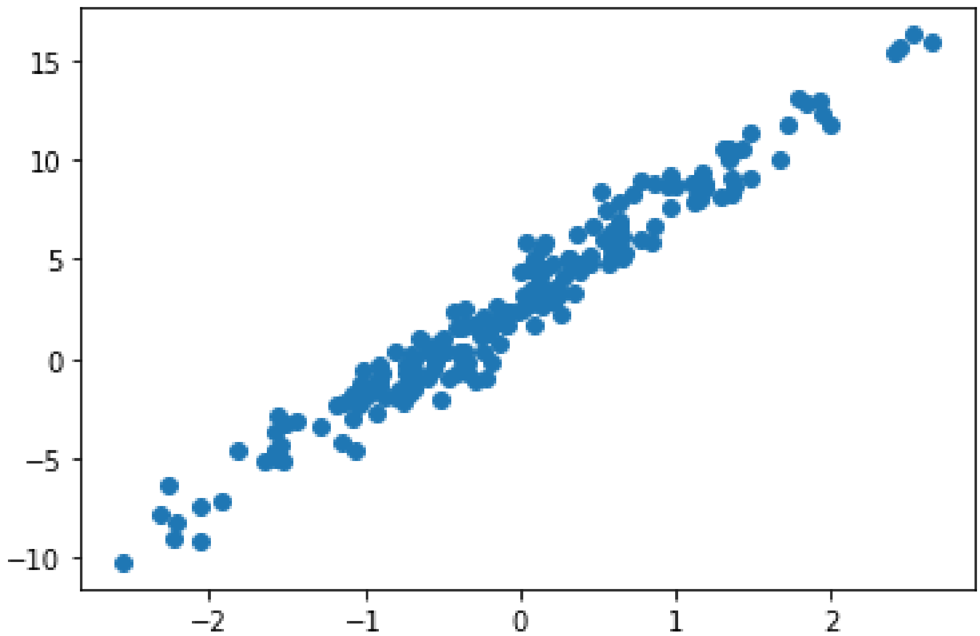
```
import matplotlib.pyplot as plt
import tensorflow as tf
%matplotlib inline
####### >>>>>填写<<<<<   设置你想要的真实斜率和截距， 以及数据点(50
-200) >>>>>填写<<<<< ############
true_w = 5
true_b = 3
num_samples = 200

#########################################################
######################################


# 初始化随机数据
X = tf.random.normal(shape=[num_samples, 1]).numpy()
noise = tf.random.normal(shape=[num_samples, 1]).numpy()

####### >>>>>填写<<<<< 在下述函数尾端，添加噪音项，注意删除下划线
 >>>>>填写<<<<<  #################################
y = X * true_w + true_b + noise
########################################

plt.scatter(X, y)
```

Out[4]:

```
<matplotlib.collections.PathCollection at 0x140313dd8>
```

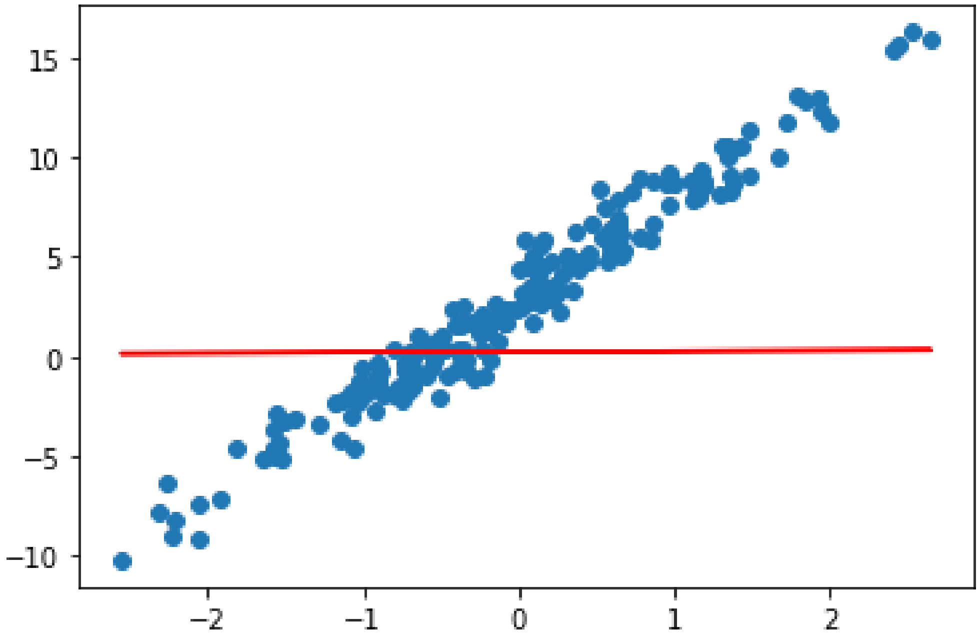



```
W = tf.Variable(tf.random.uniform([1]))  # 随机初始化参数
b = tf.Variable(tf.random.uniform([1]))
```

```python
# >>>>>填写<<<<<  定义一个随机一元一次线性函数  y = W * x + b
 >>>>>填写<<<<<  ##
# 思考 这里的x 为什么是小写？ #
def random_line(x):

    y = W * x + b
    return y

plt.scatter(X, y)
plt.plot(X, random_line(X), c='r')
```

Out[6]:

```
[<matplotlib.lines.Line2D at 0x1403c51d0>]
```




```python
def loss_fn( x, y):
    y_ = random_line(x)
    return tf.reduce_mean(tf.square(y_ - y))
### >>>>>填写<<<<< 通过改变epochs 的值，推荐起始10，  和learning rate
 学习率 推荐0.1 起始 观察梯度下降学习的线性函数w b 的值，以及loss 函数
的变化>>>>>填写<<<<<   ######

EPOCHS = 50
LEARNING_RATE = 0.01
##########################################################
##########################################################
##########################
```

```
for epoch in range(EPOCHS):   # 迭代次数
    with tf.GradientTape() as tape:   # 追踪梯度
        loss = loss_fn(X, y)   # 计算损失
    dW, db = tape.gradient(loss, [W, b])   # 计算梯度
    W.assign_sub(LEARNING_RATE * dW)   # 更新梯度
    b.assign_sub(LEARNING_RATE * db)
    # 输出计算过程
    print('Epoch [{}/{}], loss [{:.3f}], W is [{:.3f}], b is [{:.3f}]'.format(epoc
h, EPOCHS, loss,

float(W.numpy()),

float(b.numpy())))
Epoch [0/50], loss [33.248], W is [0.134], b is [0.296]
Epoch [1/50], loss [31.960], W is [0.232], b is [0.349]
Epoch [2/50], loss [30.724], W is [0.328], b is [0.401]
Epoch [3/50], loss [29.538], W is [0.422], b is [0.453]
Epoch [4/50], loss [28.398], W is [0.514], b is [0.503]
Epoch [5/50], loss [27.305], W is [0.605], b is [0.552]
Epoch [6/50], loss [26.255], W is [0.694], b is [0.600]
Epoch [7/50], loss [25.247], W is [0.780], b is [0.648]
Epoch [8/50], loss [24.279], W is [0.865], b is [0.694]
Epoch [9/50], loss [23.350], W is [0.949], b is [0.739]
Epoch [10/50], loss [22.458], W is [1.031], b is [0.784]
Epoch [11/50], loss [21.601], W is [1.111], b is [0.827]
Epoch [12/50], loss [20.779], W is [1.189], b is [0.870]
Epoch [13/50], loss [19.990], W is [1.266], b is [0.912]
Epoch [14/50], loss [19.232], W is [1.341], b is [0.953]
Epoch [15/50], loss [18.505], W is [1.415], b is [0.993]
Epoch [16/50], loss [17.806], W is [1.487], b is [1.032]
Epoch [17/50], loss [17.136], W is [1.558], b is [1.071]
Epoch [18/50], loss [16.492], W is [1.628], b is [1.109]
Epoch [19/50], loss [15.874], W is [1.696], b is [1.146]
Epoch [20/50], loss [15.281], W is [1.762], b is [1.182]
Epoch [21/50], loss [14.711], W is [1.827], b is [1.217]
Epoch [22/50], loss [14.164], W is [1.891], b is [1.252]
Epoch [23/50], loss [13.639], W is [1.954], b is [1.286]
Epoch [24/50], loss [13.135], W is [2.015], b is [1.320]
Epoch [25/50], loss [12.651], W is [2.076], b is [1.352]
Epoch [26/50], loss [12.186], W is [2.135], b is [1.385]
Epoch [27/50], loss [11.740], W is [2.192], b is [1.416]
Epoch [28/50], loss [11.312], W is [2.249], b is [1.447]
Epoch [29/50], loss [10.901], W is [2.304], b is [1.477]
Epoch [30/50], loss [10.506], W is [2.359], b is [1.507]
```

```
Epoch [31/50], loss [10.127], W is [2.412], b is [1.536]
Epoch [32/50], loss [9.764], W is [2.464], b is [1.564]
Epoch [33/50], loss [9.414], W is [2.515], b is [1.592]
Epoch [34/50], loss [9.079], W is [2.565], b is [1.619]
Epoch [35/50], loss [8.757], W is [2.614], b is [1.646]
Epoch [36/50], loss [8.448], W is [2.663], b is [1.672]
Epoch [37/50], loss [8.151], W is [2.710], b is [1.698]
Epoch [38/50], loss [7.867], W is [2.756], b is [1.723]
Epoch [39/50], loss [7.593], W is [2.801], b is [1.747]
Epoch [40/50], loss [7.331], W is [2.845], b is [1.771]
Epoch [41/50], loss [7.079], W is [2.889], b is [1.795]
Epoch [42/50], loss [6.837], W is [2.931], b is [1.818]
Epoch [43/50], loss [6.604], W is [2.973], b is [1.841]
Epoch [44/50], loss [6.381], W is [3.014], b is [1.863]
Epoch [45/50], loss [6.167], W is [3.054], b is [1.885]
Epoch [46/50], loss [5.962], W is [3.093], b is [1.906]
Epoch [47/50], loss [5.764], W is [3.132], b is [1.927]
Epoch [48/50], loss [5.575], W is [3.169], b is [1.948]
Epoch [49/50], loss [5.393], W is [3.206], b is [1.968]
```