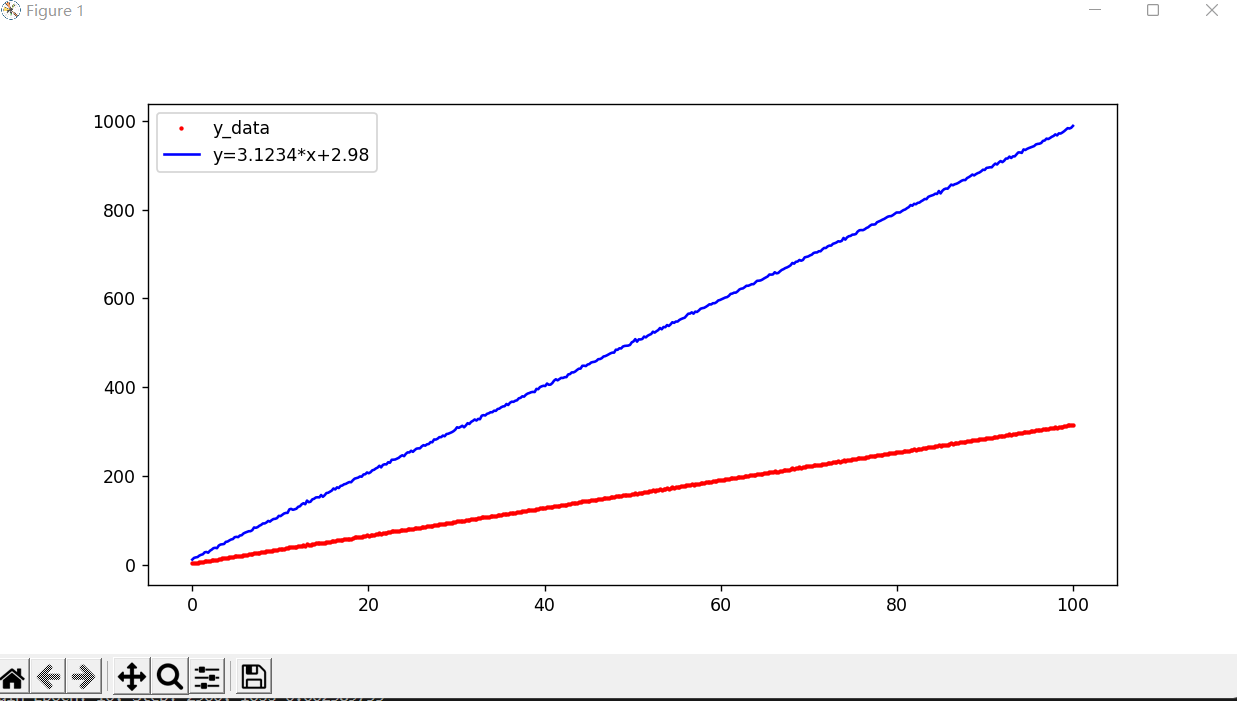
生成数据的散点图：



代码源码：

import numpy as np

import tensorflow as tf

tf.compat.v1.disable\_eager\_execution()

import matplotlib.pyplot as plt

x\_data = np.linspace(0,100,500)

y\_data = 3.1234\* x\_data + 2.98 + np.random.randn(500) \* 0.5

plt.figure(figsize=[10,5])

plt.plot(x\_data,y\_data,'r.',markersize=3)

plt.plot(x\_data,y\_data \*3.1234 + 2.98 ,'b-')

plt.legend(['y\_data','y=3.1234\*x+2.98'])

# plt.show()

x = tf.compat.v1.placeholder('float',name='x')

y = tf.compat.v1.placeholder('float',name='y')

w = tf.compat.v1.Variable(1.0,name='w0')

b = tf.compat.v1.Variable(1.0,name='b0')

def model(x,w,b):

return x\*w+b

pred = model(x,w,b)

#训练模型

train\_epochs = 10

learning\_rate = 0.0001

display\_step = 20

loss\_function = tf.reduce\_mean(tf.square(y - pred))

optimizer = tf.compat.v1.train.GradientDescentOptimizer(learning\_rate).minimize(loss\_function)

sess = tf.compat.v1.Session()

init = tf.compat.v1.global\_variables\_initializer()

loss\_list = []

step = 0

sess.run(init)

for epoch in range(train\_epochs):

for xs, ys in zip(x\_data, y\_data):

\_, loss = sess.run([optimizer, loss\_function], feed\_dict={x: xs, y: ys})

loss\_list.append(loss)

step += 1

if step % display\_step == 0:

print(f'Train Epoch: {epoch+1:02d}, Step: {step:03d}, loss={loss:.9f}')

x\_test = 5.79

y\_hat = sess.run(model(x\_test, w, b))

y\_target = 3.1234 \* x\_test + 2.98

print(f'当x=5.79时，目标值为{y\_target}，模型预测值为{y\_hat}')

logdir =r'C:/Users/19085/my\_log\_dir'

tf.compat.v1.summary.FileWriter(logdir, tf.compat.v1.get\_default\_graph()).close()

通过Tensorboard显示构建的计算图：

