

Unit 6 Softmax Regression

Part 03 Introduction to TensorFlow

TFIP-AI Artificial Neural Networks and Deep Learning

Motivating problem

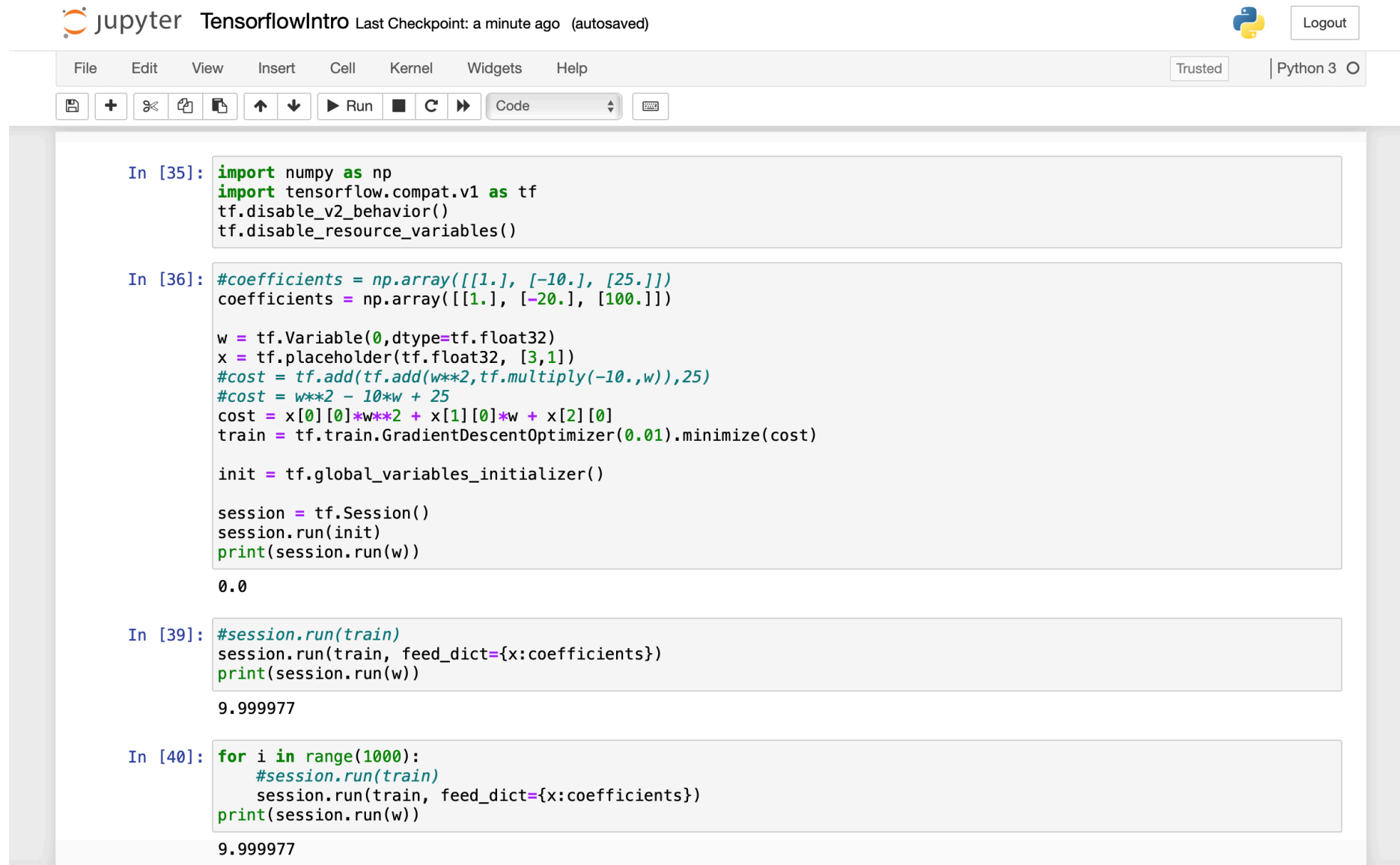
- Cost:

$$\begin{aligned}\mathbf{J}(w) &= w^2 - 10w + 25 \\ &= (w - 5)^2\end{aligned}$$

$$w = 5$$

$$\mathbf{J}(w, b)$$

An Example with TensorFlow version 1



The image shows a Jupyter Notebook interface with the title "TensorflowIntro" and a status bar indicating "Last Checkpoint: a minute ago (autosaved)". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for saving, adding cells, and running code. The notebook contains four code cells:

```
In [35]: import numpy as np
import tensorflow.compat.v1 as tf
tf.disable_v2_behavior()
tf.disable_resource_variables()

In [36]: #coefficients = np.array([[1.], [-10.], [25.]])
coefficients = np.array([[1.], [-20.], [100.]])

w = tf.Variable(0, dtype=tf.float32)
x = tf.placeholder(tf.float32, [3, 1])
#cost = tf.add(tf.add(w**2, tf.multiply(-10., w)), 25)
#cost = w**2 - 10*w + 25
cost = x[0][0]*w**2 + x[1][0]*w + x[2][0]
train = tf.train.GradientDescentOptimizer(0.01).minimize(cost)

init = tf.global_variables_initializer()

session = tf.Session()
session.run(init)
print(session.run(w))

0.0

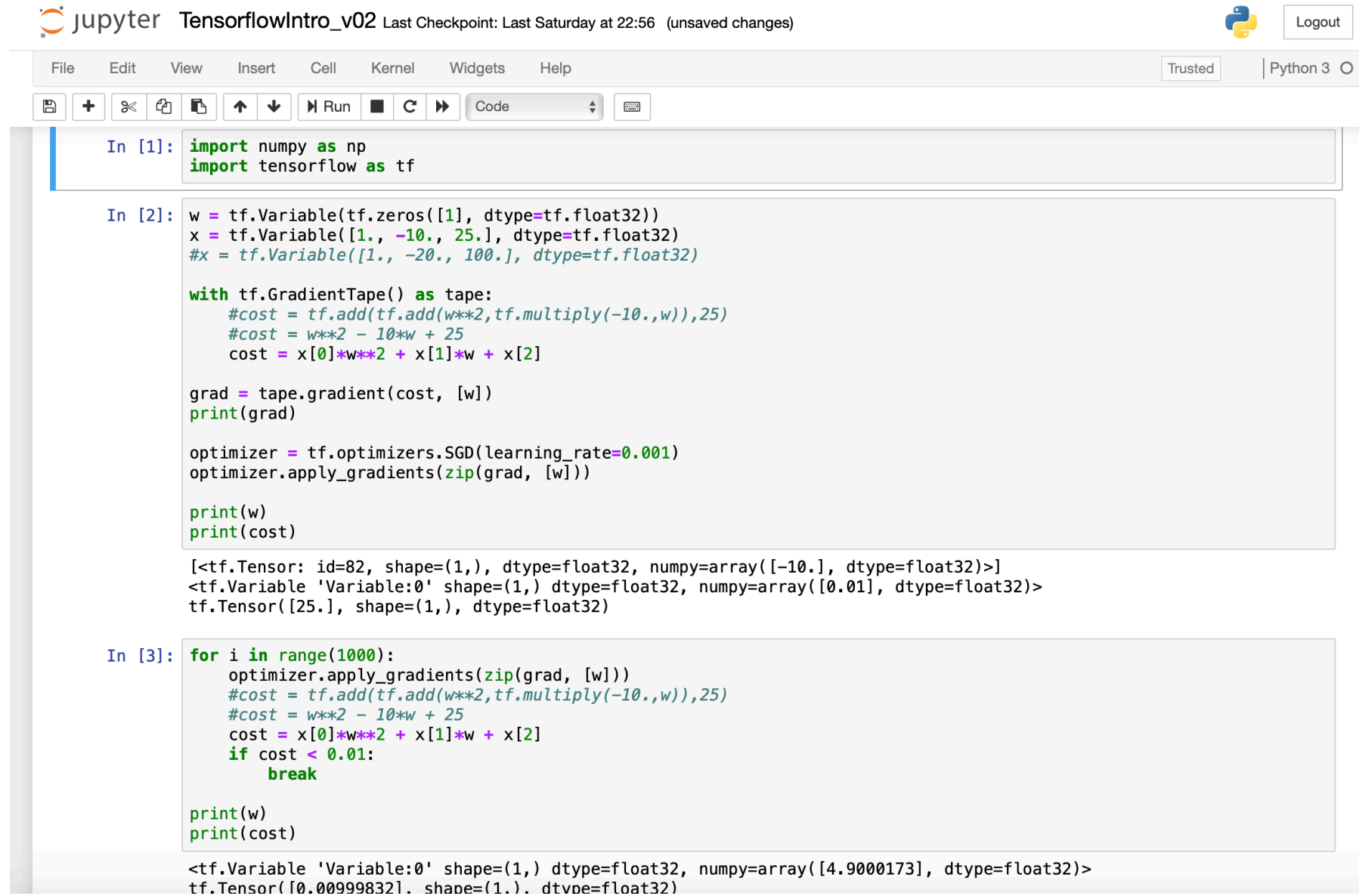
In [39]: #session.run(train)
session.run(train, feed_dict={x: coefficients})
print(session.run(w))

9.999977

In [40]: for i in range(1000):
#session.run(train)
    session.run(train, feed_dict={x: coefficients})
print(session.run(w))

9.999977
```

An Example with TensorFlow version 2



The image shows a Jupyter Notebook interface with the title "TensorflowIntro_v02". The top bar includes the Jupyter logo, the notebook name, the last checkpoint time ("Last Saturday at 22:56"), and a "Logout" button. Below the top bar is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". To the right of the menu bar are "Trusted" and "Python 3" buttons. Below the menu bar is a toolbar with icons for saving, adding cells, undo, redo, and running code. The notebook contains three code cells. The first cell imports numpy and tensorflow. The second cell defines variables w and x, calculates the cost function, and prints the gradient and cost. The third cell runs an optimization loop for 1000 iterations and prints the final w and cost. The output of the second cell shows the initial values of w and x, and the output of the third cell shows the final values of w and cost after 1000 iterations.

```
In [1]: import numpy as np
import tensorflow as tf

In [2]: w = tf.Variable(tf.zeros([1], dtype=tf.float32))
x = tf.Variable([1., -10., 25.], dtype=tf.float32)
#x = tf.Variable([1., -20., 100.], dtype=tf.float32)

with tf.GradientTape() as tape:
    #cost = tf.add(tf.add(w**2, tf.multiply(-10., w)), 25)
    #cost = w**2 - 10*w + 25
    cost = x[0]*w**2 + x[1]*w + x[2]

grad = tape.gradient(cost, [w])
print(grad)

optimizer = tf.optimizers.SGD(learning_rate=0.001)
optimizer.apply_gradients(zip(grad, [w]))

print(w)
print(cost)

[<tf.Tensor: id=82, shape=(1,), dtype=float32, numpy=array([-10.], dtype=float32)>]
<tf.Variable 'Variable:0' shape=(1,) dtype=float32, numpy=array([0.01], dtype=float32)>
tf.Tensor([25.], shape=(1,), dtype=float32)

In [3]: for i in range(1000):
optimizer.apply_gradients(zip(grad, [w]))
#cost = tf.add(tf.add(w**2, tf.multiply(-10., w)), 25)
#cost = w**2 - 10*w + 25
cost = x[0]*w**2 + x[1]*w + x[2]
if cost < 0.01:
    break

print(w)
print(cost)

<tf.Variable 'Variable:0' shape=(1,) dtype=float32, numpy=array([4.9000173], dtype=float32)>
tf.Tensor([0.00999832], shape=(1,), dtype=float32)
```

Another Example with TensorFlow version 2

jupyter TensorflowIntro_v02_01 Last Checkpoint: Last Sunday at 10:28 (autosaved)



Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted

Python 3

Code

```
In [2]: import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt

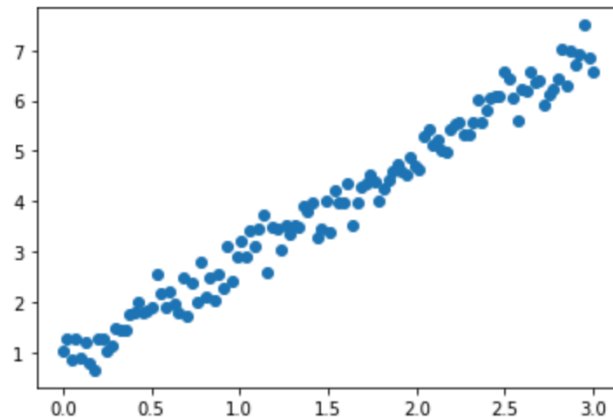
print(tf.__version__)
```

2.0.0


```
In [3]: # actual weight = 2 and actual bias = 0.9
x = np.linspace(0, 3, 120)
y = 2 * x + 0.9 + np.random.randn(*x.shape) * 0.3

plt.scatter(x,y, label="input data set")
```

Out[3]: <matplotlib.collections.PathCollection at 0x7fbd9e5465d0>



Another Example with TensorFlow version 2 cont...

jupyter TensorflowIntro_v02_01 Last Checkpoint: Last Sunday at 10:28 (autosaved)  Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
In [4]: class LinearModel:
        def __call__(self, x):
            return self.Weight * x + self.Bias

        def __init__(self):
            self.Weight = tf.Variable(11.0)
            self.Bias = tf.Variable(12.0)

        def loss(y, pred):
            return tf.reduce_mean(tf.square(y - pred))


        def train(linear_model, x, y, lr=0.12):
            with tf.GradientTape() as t:
                current_loss = loss(y, linear_model(x))

            lr_weight, lr_bias = t.gradient(current_loss, [linear_model.Weight, linear_model.Bias])
            linear_model.Weight.assign_sub(lr * lr_weight)
            linear_model.Bias.assign_sub(lr * lr_bias)

In [5]: linear_model = LinearModel()
Weights, Biases = [], []
epochs = 80
for epoch_count in range(epochs):
    Weights.append(linear_model.Weight.numpy())
    Biases.append(linear_model.Bias.numpy())
    real_loss = loss(y, linear_model(x))
    train(linear_model, x, y, lr=0.12)
    print(f"Epoch count {epoch_count}: Loss value: {real_loss.numpy()}")

Epoch count 0: Loss value: 666.9447021484375
Epoch count 1: Loss value: 10.690195083618164
Epoch count 2: Loss value: 5.408223628997803
Epoch count 3: Loss value: 4.878205299377441
Epoch count 4: Loss value: 4.428596496582031
Epoch count 5: Loss value: 4.021341323852539
Epoch count 6: Loss value: 3.652250051498413
Epoch count 7: Loss value: 3.3177449703216553
```


Another Example with TensorFlow version 2 cont...


jupyter TensorflowIntro_v02_01 Last Checkpoint: Last Sunday at 10:28 (autosaved)  Logout













File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
Epoch count 7: Loss value: 3.5177449705210555
Epoch count 8: Loss value: 3.014585018157959
Epoch count 9: Loss value: 2.7398340702056885
Epoch count 10: Loss value: 2.490828037261963
Epoch count 11: Loss value: 2.265155792236328
Epoch count 12: Loss value: 2.0606303215026855
Epoch count 13: Loss value: 1.875271201133728
Epoch count 14: Loss value: 1.7072807550430298
Epoch count 15: Loss value: 1.5550323724746704
Epoch count 16: Loss value: 1.4170507192611694
Epoch count 17: Loss value: 1.291998267173767
Epoch count 18: Loss value: 1.1786646842956543
Epoch count 19: Loss value: 1.0759508609771729
Epoch count 20: Loss value: 0.9828624725341797
Epoch count 21: Loss value: 0.8984968066215515
Epoch count 22: Loss value: 0.8220367431640625
Epoch count 23: Loss value: 0.7527414560317993
Epoch count 24: Loss value: 0.6899397373199463
Epoch count 25: Loss value: 0.6330228447914124
Epoch count 26: Loss value: 0.5814394354820251
Epoch count 27: Loss value: 0.534689724445343
Epoch count 28: Loss value: 0.49232086539268494
Epoch count 29: Loss value: 0.45392221212387085
Epoch count 30: Loss value: 0.41912174224853516
Epoch count 31: Loss value: 0.38758227229118347
Epoch count 32: Loss value: 0.3589983582496643
Epoch count 33: Loss value: 0.33309295773506165
Epoch count 34: Loss value: 0.30961498618125916
Epoch count 35: Loss value: 0.28833702206611633
Epoch count 36: Loss value: 0.26905301213264465
Epoch count 37: Loss value: 0.2515760064125061
Epoch count 38: Loss value: 0.23573674261569977
Epoch count 39: Loss value: 0.22138167917728424
Epoch count 40: Loss value: 0.20837180316448212
Epoch count 41: Loss value: 0.19658105075359344
Epoch count 42: Loss value: 0.185895174741745
Epoch count 43: Loss value: 0.17621064186096191
Epoch count 44: Loss value: 0.16743357479572296
Epoch count 45: Loss value: 0.15947897732257843
```

Another Example with TensorFlow version 2 cont...


jupyter TensorflowIntro_v02_01 Last Checkpoint: Last Sunday at 10:28 (autosaved)  Logout

File Edit View Insert Cell Kernel Widgets Help Trusted | Python 3 

           Code 

```
Epoch count 43: Loss value: 0.17621064186096191
Epoch count 44: Loss value: 0.16743357479572296
Epoch count 45: Loss value: 0.15947897732257843
Epoch count 46: Loss value: 0.15226984024047852
Epoch count 47: Loss value: 0.1457361876964569
Epoch count 48: Loss value: 0.1398148089647293
Epoch count 49: Loss value: 0.13444828987121582
Epoch count 50: Loss value: 0.1295846551656723
Epoch count 51: Loss value: 0.125176802277565
Epoch count 52: Loss value: 0.12118196487426758
Epoch count 53: Loss value: 0.1175614520907402
Epoch count 54: Loss value: 0.11428023874759674
Epoch count 55: Loss value: 0.11130647361278534
Epoch count 56: Loss value: 0.10861136019229889
Epoch count 57: Loss value: 0.10616883635520935
Epoch count 58: Loss value: 0.10395515710115433
Epoch count 59: Loss value: 0.10194893181324005
Epoch count 60: Loss value: 0.10013069957494736
Epoch count 61: Loss value: 0.09848284721374512
Epoch count 62: Loss value: 0.09698941558599472
Epoch count 63: Loss value: 0.09563592076301575
Epoch count 64: Loss value: 0.09440924972295761
Epoch count 65: Loss value: 0.0932975485920906
Epoch count 66: Loss value: 0.092289999127388
Epoch count 67: Loss value: 0.0913768783211708
Epoch count 68: Loss value: 0.09054934233427048
Epoch count 69: Loss value: 0.08979929238557816
Epoch count 70: Loss value: 0.0891195759177208
Epoch count 71: Loss value: 0.08850353211164474
Epoch count 72: Loss value: 0.08794523030519485
Epoch count 73: Loss value: 0.08743925392627716
Epoch count 74: Loss value: 0.0869806632399559
Epoch count 75: Loss value: 0.0865650624036789
Epoch count 76: Loss value: 0.08618839830160141
Epoch count 77: Loss value: 0.08584703505039215
Epoch count 78: Loss value: 0.08553765714168549
Epoch count 79: Loss value: 0.08525727689266205
```


Another Example with TensorFlow version 2 cont...

jupyter TensorflowIntro_v02_01 Last Checkpoint: Last Sunday at 10:28 (autosaved)  Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Code

```
Epoch count 61: Loss value: 0.09848284721374512
Epoch count 62: Loss value: 0.09698941558599472
Epoch count 63: Loss value: 0.09563592076301575
Epoch count 64: Loss value: 0.09440924972295761
Epoch count 65: Loss value: 0.0932975485920906
Epoch count 66: Loss value: 0.092289999127388
Epoch count 67: Loss value: 0.0913768783211708
Epoch count 68: Loss value: 0.09054934233427048
Epoch count 69: Loss value: 0.08979929238557816
Epoch count 70: Loss value: 0.0891195759177208
Epoch count 71: Loss value: 0.08850353211164474
Epoch count 72: Loss value: 0.08794523030519485
Epoch count 73: Loss value: 0.08743925392627716
Epoch count 74: Loss value: 0.0869806632399559
Epoch count 75: Loss value: 0.0865650624036789
Epoch count 76: Loss value: 0.08618839830160141
Epoch count 77: Loss value: 0.08584703505039215
Epoch count 78: Loss value: 0.08553765714168549
Epoch count 79: Loss value: 0.08525727689266205
```

In [6]: `linear_model.Weight.numpy(), linear_model.Bias.numpy()`

Out[6]: `(2.0053532, 0.89694154)`

In [7]: `RMSE = loss(y, linear_model(x))`
`RMSE.numpy()`

Out[7]: `0.08500317`

In []: