# 01_linear_regression_using_tensorflow_homework

September 23, 2020

[1]:

```
import numpy as np

np.__version__
```

[1]:

```
'1.19.1'
```

[ ]:

```

```

[2]:

```
def AND(x1, x2) :
    x = np.array([x1, x2])
    w = np.array([0.5, 0.5])
    b = -0.7

    tmp = np.sum(w*x) + b
    if tmp <=0 :
        return 0
    else :
        return 1


print(AND(1, 1))
print(AND(1, 0))
print(AND(0, 1))
print(AND(0, 0))
```

```
1
0
0
0
```

[3]:

```
def NAND(x1, x2) :
    x = np.array([x1, x2])
    w = np.array([-0.5, -0.5])
    b = 0.7

    tmp = np.sum(w*x) + b
    if tmp <=0 :
        return 0
    else :
        return 1

print(NAND(1, 1))
print(NAND(1, 0))
print(NAND(0, 1))
print(NAND(0, 0))
```

```
0
1
1
1
```

[4]:

```
def OR(x1, x2) :
    x = np.array([x1, x2])
    w = np.array([0.5, 0.5])
    b = -0.2

    tmp = np.sum(w*x) + b
    if tmp <= 0:
        return 0
    else :
        return 1
print(OR(1, 1))
print(OR(1, 0))
print(OR(0, 1))
print(OR(0, 0))
```

```
1
1
1
0
```

[5]:

```
def XOR(x1, x2) :
    s1 = NAND(x1, x2)
    s2 = OR(x1, x2)
    y = AND(s1, s2)
    return y

print(XOR(1, 1))
print(XOR(1, 0))
print(XOR(0, 1))
print(XOR(0, 0))
```

```
0
1
1
0
```

[6]:

```
#import tensorflow as tf
import tensorflow.compat.v1 as tf
tf.disable_v2_behavior()

import numpy as np
import matplotlib.pyplot as plt
```

```
WARNING:tensorflow:From C:\Users\kyeong min\anaconda3\envs\tensorflow\lib\site-
packages\tensorflow_core\python\compat\v2_compat.py:88:
disable_resource_variables (from tensorflow.python.ops.variable_scope) is
deprecated and will be removed in a future version.
Instructions for updating:
non-resource variables are not supported in the long term
```

[7]:

```
# x_train = [1, 2, 3]

# y_train = [2+0.1, 4-0.3, 6+0.15] #    noise

# y_train = np.multiply(x_train, 2)


# y_train = [3, 5, 7]
```

[8]:

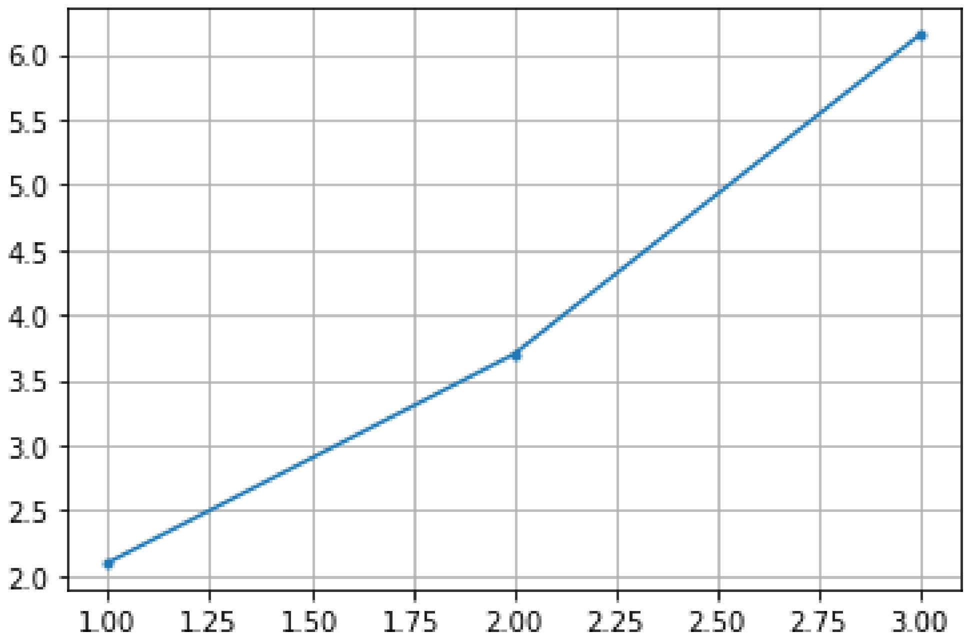
```
# x_train = np.arange(1.0, 5.0, 0.1)
# y_train = np.log(x_train)
# b = np.random.randn()

# y_train = y_train+b


x_train = [1, 2, 3]

y_train = [2+0.1, 4-0.3, 6+0.15] #    noise

y_train
```

[8]:

```
[2.1, 3.7, 6.15]
```

[9]:

```
plt.plot(x_train, y_train, '.-')
plt.grid()
```



[10]:
```
w0 = 7.0
b0 = 5.0

w0 = tf.Variable(tf.random_normal([1]), name = 'weight')
b0 = tf.Variable(tf.random_normal([1]), name = 'bias')
```

[11]:
```
hypothesis = x_train * w0 + b0
hypothesis
```

[11]:
```
<tf.Tensor 'add:0' shape=(3,) dtype=float32>
```

[12]:
```
loss = tf.reduce_mean(tf.square(hypothesis - y_train))
loss
```

[12]:
```
<tf.Tensor 'Mean:0' shape=() dtype=float32>
```

[13]:
```
optimizer = tf.train.GradientDescentOptimizer(learning_rate = 0.01)
train = optimizer.minimize(loss)
train
```

[13]:
```
<tf.Operation 'GradientDescent' type=NoOp>
```

[14]:
```
sess = tf.Session()
sess
```

[14]:
```
<tensorflow.python.client.session.Session at 0x15dc248a108>
```

[15]:

```
sess.run(tf.global_variables_initializer())
```

[17]:

```
nb_epoch = 10001
vloss = []
vb = []
vw = []

for step in range(nb_epoch) :
    sess.run(train)

    if step % 200 == 0 :
        w1 = sess.run(w0)[0]
        b1 = sess.run(b0)[0]
        loss1 = sess.run(loss)
        vb.append(b1)
        vw.append(w1)


        vloss.append(loss1)
        print(step, '\t', loss1, w1, b1)
```

```
0        25.858393 -0.2852242 -0.16428608
200      0.07259211 1.8157694 0.40896356
400      0.052531093 1.8957083 0.2272437
600      0.04487081 1.9451059 0.11495186
800      0.041945796 1.9756303 0.045562427
1000     0.04082884 1.9944925 0.0026841455
1200     0.040402357 2.006148 -0.02381174
1400     0.04023951 2.0133505 -0.040184755
1600     0.040177274 2.0178013 -0.050302256
1800     0.040153526 2.0205512 -0.05655402
2000     0.040144462 2.0222504 -0.06041666
2200     0.040141057 2.0233 -0.06280288
2400     0.040139653 2.023949 -0.0642769
2600     0.040139172 2.0243497 -0.06518767
2800     0.040138967 2.0245962 -0.06575075
3000     0.04013895 2.0247505 -0.066099584
3200     0.0401389 2.0248456 -0.066315204
3400     0.040138856 2.0249023 -0.06644599
3600     0.04013888 2.0249414 -0.06653266
3800     0.04013894 2.0249631 -0.06658347
4000     0.040138867 2.0249755 -0.066612415
4200     0.040138856 2.0249827 -0.0666289
4400     0.040138904 2.0249867 -0.06663825
4600     0.040138926 2.024989 -0.06664352
```

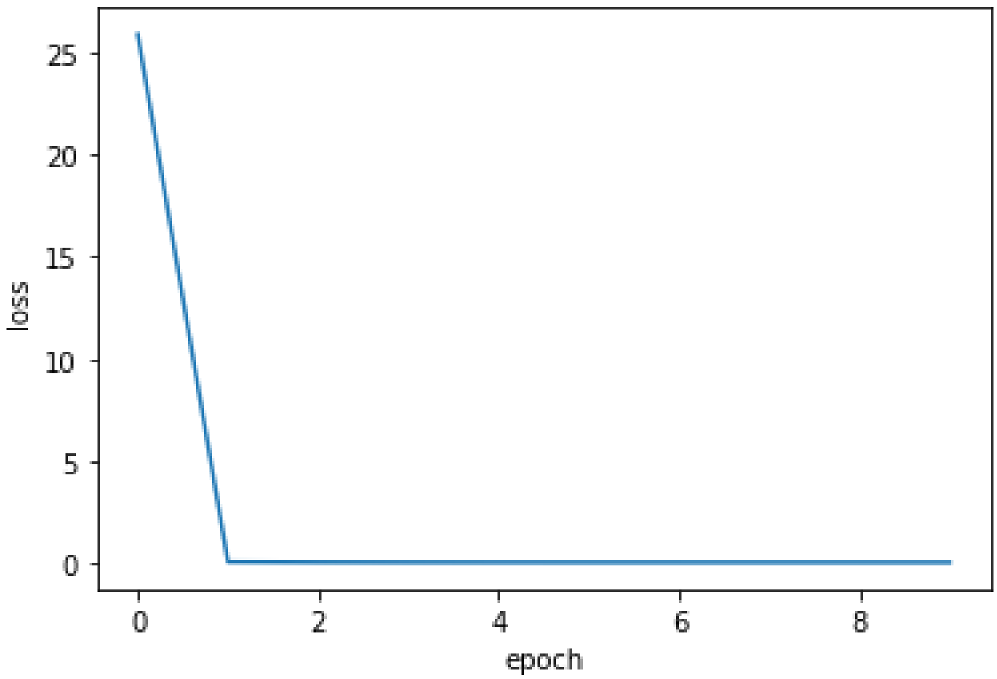
```
4800     0.04013887 2.02499 -0.06664646
5000     0.040138904 2.0249908 -0.06664799
5200     0.040138904 2.0249908 -0.06664799
5400     0.040138904 2.0249908 -0.06664799
5600     0.040138904 2.0249908 -0.06664799
5800     0.040138904 2.0249908 -0.06664799
6000     0.040138904 2.0249908 -0.06664799
6200     0.040138904 2.0249908 -0.06664799
6400     0.040138904 2.0249908 -0.06664799
6600     0.040138904 2.0249908 -0.06664799
6800     0.040138904 2.0249908 -0.06664799
7000     0.040138904 2.0249908 -0.06664799
7200     0.040138904 2.0249908 -0.06664799
7400     0.040138904 2.0249908 -0.06664799
7600     0.040138904 2.0249908 -0.06664799
7800     0.040138904 2.0249908 -0.06664799
8000     0.040138904 2.0249908 -0.06664799
8200     0.040138904 2.0249908 -0.06664799
8400     0.040138904 2.0249908 -0.06664799
8600     0.040138904 2.0249908 -0.06664799
8800     0.040138904 2.0249908 -0.06664799
9000     0.040138904 2.0249908 -0.06664799
9200     0.040138904 2.0249908 -0.06664799
9400     0.040138904 2.0249908 -0.06664799
9600     0.040138904 2.0249908 -0.06664799
9800     0.040138904 2.0249908 -0.06664799
10000    0.040138904 2.0249908 -0.06664799
```

[18]:

```
plt.plot(vloss[:10])
plt.xlabel("epoch")
plt.ylabel("loss")
```

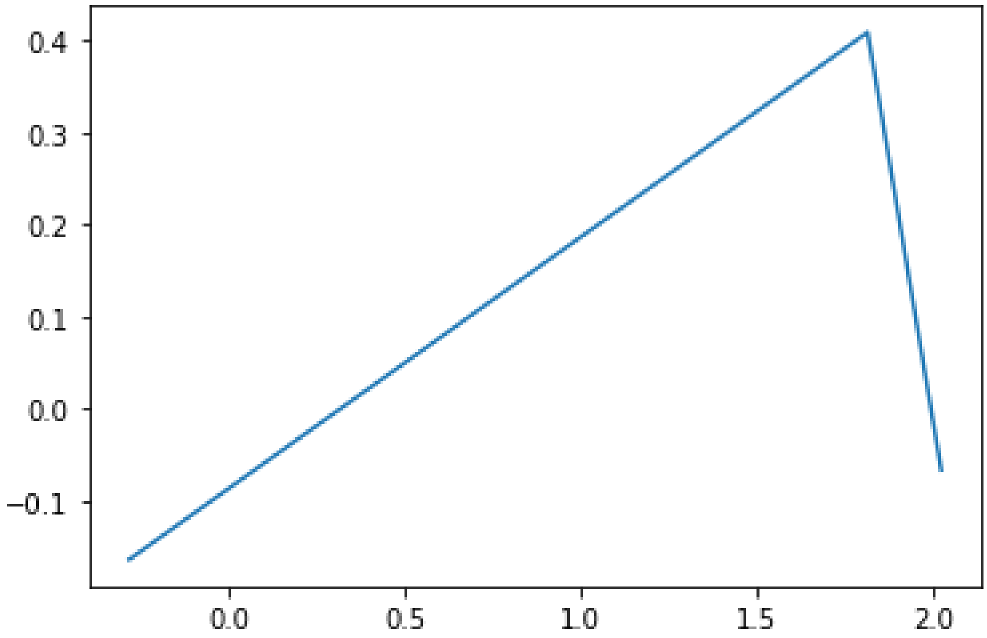
[18]:

```
Text(0, 0.5, 'loss')
```



```
[19]: plt.plot(vw, vb)
```

```
[19]: [<matplotlib.lines.Line2D at 0x15dc24f2508>]
```



[21]:

```
w1 = sess.run(w0)[0]
b1 = sess.run(b0)[0]
print(w1, b1)
```

```
2.0249908 -0.06664799
```

[22]:

```
str1 = 'y = ' + str(w1) +'x + ' + str(b1)
print(str1)
```

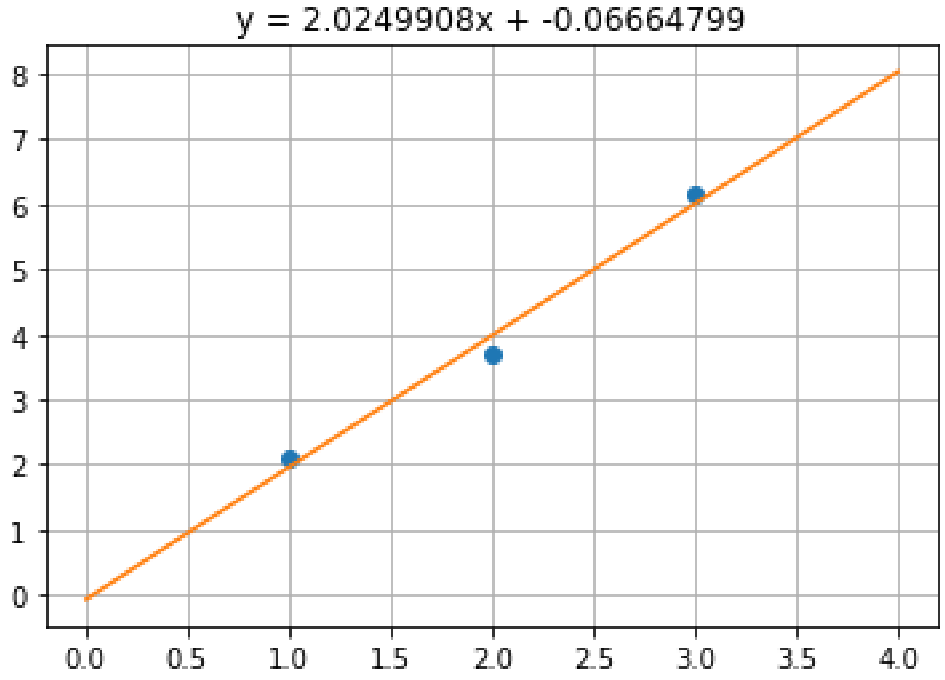
```
y = 2.0249908x + -0.06664799
```

[23]:

```
plt.figure(figsize = (6, 4))
plt.plot(x_train, y_train, "o")

x1 = np.linspace(np.min(x_train)-1, np.max(x_train)+1)
y1 = w1*x1 + b1
plt.plot(x1, y1)
plt.grid()

plt.title(str1)
```

[23]:

```
Text(0.5, 1.0, 'y = 2.0249908x + -0.06664799')
```


y = 2.0249908x + -0.06664799
8
7
6
5
4
3
2
1
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0