Linear Regression

OBJECTIVE: Understand and practice linear regression.

- Very important !

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

X and Y data

```
x_train = [1, 2, 3, 4, 5]
y_train = [2+0.1+3, 4-0.3+3, 6+0.15+3, 8+0.2+3, 10-0.2+3] #  Add some noise
```

Initialization

```
print(tf.Variable(w0 * tf.ones([1]), name='weight'))
```

```
<tf.Variable 'weight_1:0' shape=(1,) dtype=float32_ref>
```

```
#W = tf.Variable(tf.random_normal([1]), name='weight')
#b = tf.Variable(tf.random_normal([1]), name='bias')
w0 = 4000.0;
b0 = 5.0;

W = tf.Variable(w0 * tf.ones([1]), name='weight') # tf.ones([1]) => [1]
b = tf.Variable(b0 * tf.ones([1]), name='bias')
# tf.ones([3,3]) => [1,1,1],[1,1,1],[1,1,1]
# tf.zeros([3,3]) => [0,0,0],[0,0,0],[0,0,0]
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/python/framework/c
Instructions for updating:
Colocations handled automatically by placer.
```

Our hypothesis XW+b

```
hypothesis = x_train * W + b
```

cost/loss function

```
cost = tf.reduce_mean(tf.square(hypothesis - y_train))
```

Optimizer

```
optimizer = tf.train.GradientDescentOptimizer(learning_rate=0.01)
train = optimizer.minimize(cost)
```

Launch the graph in a session

```
sess = tf.Session()
```

Initializes global variables in the graph.

```
sess.run(tf.global_variables_initializer()) # = tf.Session().run(tf.global_variables_initializer())
```

```
vw = [] # vector weight
vb = [] # vector bias

for step in range(4001):
    sess.run(train)

    w1 = sess.run(W)[0] # slope
    b1 = sess.run(b)[0] # bias
    vw.append(w1)
    vb.append(b1)

    if step % 100 == 0:
        print(step, sess.run(cost), w1, b1)
```

Complete training

```
# sess.run(train)

w1 = sess.run(W)[0] # slope
b1 = sess.run(b)[0] # bias
str1 = 'y = ' + str(w1) +'x + ' + str(b1)
print(w1, b1)
print(str1)
```



```
plt.figure(1)
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)
```


위에서 bias에 3을 추가해준 결과 3을 찾아낸 모습이다.

```
plt.plot(vw)
```

```
plt.plot(vb)
```