

# Linear Regression

## Open-book Quiz (Team)

|                  |             |
|------------------|-------------|
| Member #1 Name : | Student ID: |
| Member #2 Name : | Student ID: |
| Member #3 Name : | Student ID: |

- Briefly describe the role or meaning of the code in each of the blanks below.
- 아래의 각 빈칸에 위 부분의 코드에 대한 역할 또는 의미를 간략하게 기술하세요

In [1]:

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

In [2]:

```
x_train = [1, 2, 3, 4, 5]

#y_train = [2, 4, 6, 8, 10]
y_train = [2+0.1+3, 4-0.3+3, 6+0.15+3, 8+0.1+3, 10-0.12+3]# Add some noise
```

>>

In [3]:

```
w0 = 7.0;
b0 = 5.0;

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

>>

In [4]:

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

>>

In [5]:

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

```
>>
```

```
In [6]:
```

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

```
>>
```

## Launch the graph in a session

```
In [7]:
```

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

Initializes global variables in the graph.

```
In [8]:
```

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

```
In [9]:
```

```
vw=[] # weights
vb=[] # bias
```

```
>>
```

```
In [10]:
```

```
for step in range(1001):
    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)
```

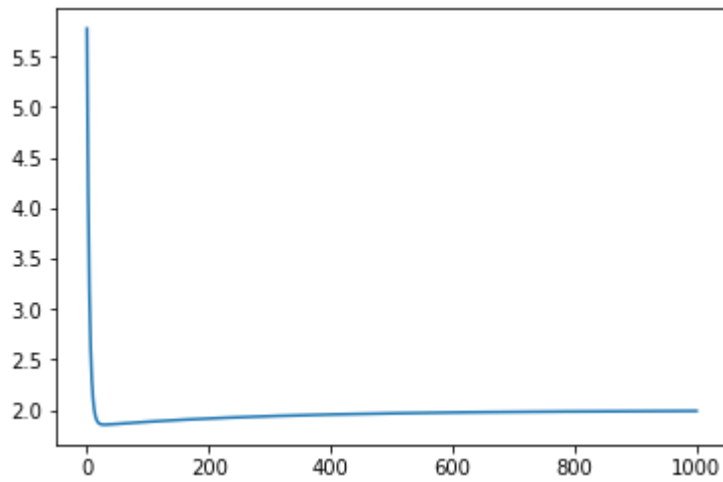
```
0 197.93019 5.7790003 4.65972
100 0.059207488 1.8834463 3.4043543
200 0.044419073 1.9157813 3.2876153
300 0.03690723 1.9388266 3.2044141
400 0.03309139 1.9552515 3.1451154
500 0.031153206 1.9669574 3.1028528
600 0.03016856 1.9753007 3.0727308
700 0.02966839 1.9812474 3.0512617
800 0.029414233 1.9854856 3.0359604
900 0.029285207 1.9885062 3.025055
1000 0.029219672 1.9906595 3.017281
```

In [11]:

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

Out[11]:

[<matplotlib.lines.Line2D at 0x129fa3390>]



## Complete training

In [12]:

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

```
1.9906595 3.017281
y = 1.9906595x + 3.017281
```

>>

In [13]:

```
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)
```

Out[13]:

Text(0.5, 1.0, 'y = 1.9906595x + 3.017281')

