# Linear Regression

드라이브에서 찾기

실습 모드에서 열기

새 Python 3 노트

새 Python 2 노트

노트 열기... Ctrl+O

노트 업로드...

이름 바꾸기...

휴지통으로 이동

드라이브에 사본 저장...

GitHub Gist로 사본 저장...

GitHub에 사본 저장...

저장 Ctrl+S

버전 저장 및 고정 Ctrl+M S

업데이트 기록

.ipynb 다운로드

.py 다운로드

드라이브 미리보기 업데이트

인쇄 Ctrl+P

Open-bo

Mem
Mem
Mem

- B ach of the blanks below.
- O 미를 간략하게 기술하세요

```
import t
import n
import m
```

```
x_train

#y_train
y_train                                    +3]#  Add some noise
```

>>

```
w0 = 7.0
b0 = 5.0

W = tf.V
b = tf.V
```

>>

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

>>

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

>>

```
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())
```

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

>>

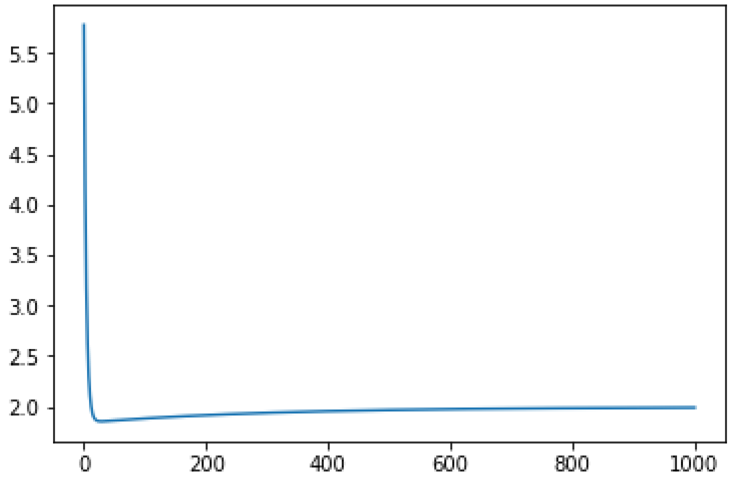
```
for step in range(1001):
    sess.run(train)
    w1 = sess.run(W)[0] # slope
    b1 = sess.run(b)[0] # bias
    vw.append(w1)
    vb.append(b1)

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

```
0 197.93013 5.779 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
```

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

```
[<matplotlib.lines.Line2D at 0x7f889c3f2240>]
```



### Complete training

```
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)
```