

▼ Linear Regression

▼ OBJECTIVE: 실습 모드에서 열기 session.

• V

import tensorflow as tf
import random
import numpy as np

X and Y

x_train

#y_train
y_train

Initialize

#W = tf.Variable([0.1, 0.1], dtype=tf.float32)
#b = tf.Variable([0.1], dtype=tf.float32)
w0 = 7.0
b0 = 5.0

W = tf.Variable([0.1, 0.1], dtype=tf.float32)
b = tf.Variable([0.1], dtype=tf.float32)

Our hypothesis

hypothesis

cost/loss

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... +3]# Add some noise
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

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())
```

```
for step in range(20):  
    if step % 3 == 1:  
        print(step)
```



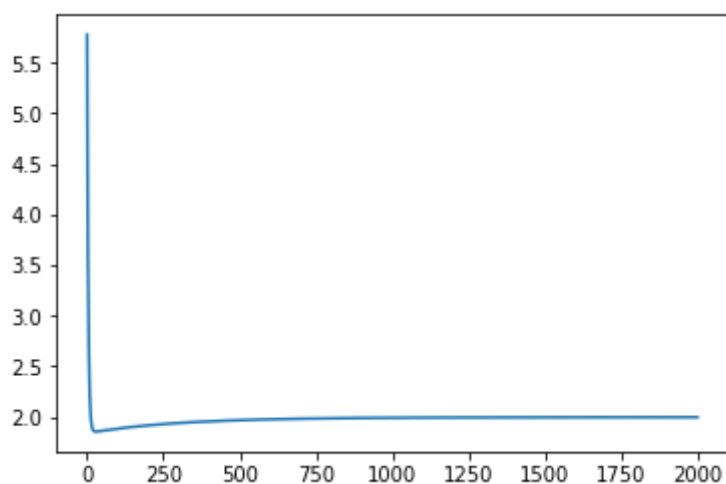
1
4
7
10
13
16
19

```
vw=[]  
vb=[]  
for step in range(2001):  
    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)
```




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

[<matplotlib.lines.Line2D at 0x7fc4bcabb048>]



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)
```



