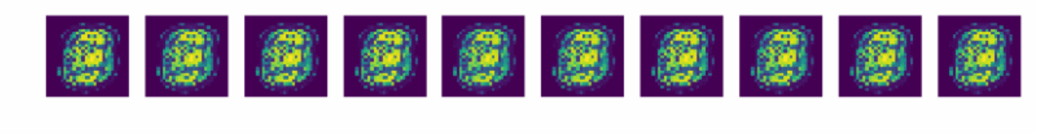
Regression

July 6, 2017

Seung-Chan Kim, Ph. D



All my great stuff, I learned outside of school

http://www.cnbc.com/2016/04/21/steve-wozniak-school-is-not-enough-go-beyond-it.html

**“There will be
this symbiosis
between humans
and machines,
in the same sense
that humans need
other humans.”**

*symbiosis (서로 다른 생물체 간의) 공생(共生)

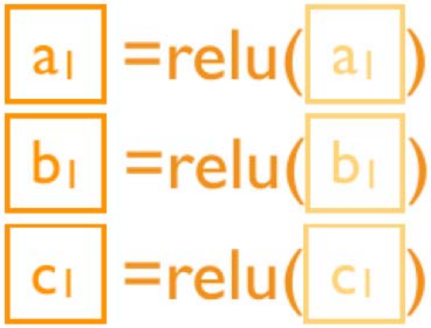
Anthes, G. (2017). Artificial intelligence poised to ride a new wave. *Communications of the ACM*, *60*(7), 19-21.
http://dl.acm.org/citation.cfm?id=3088342 (교내 접속 가능)

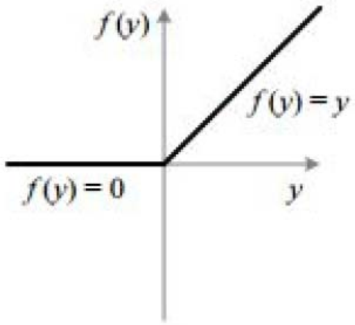
- 1. 머신러닝 개론 및 주요 개념의 이해. Tensorflow 시스템 설치 및 환경설정 (7/4 화)
- **2. Tensorflow 에 익숙해지기 실습 및 Regression의 이해 (7/6 목)**
- 3. Neural Network 이해 및 tensorflow 를 이용한 구현 (7/11 화)
- 4. 이미지 분류 이해 및 Tensorflow를 이용한 구현 (7/13 목)

Tensor manipulation : 텐서 조작해보기 (실습)

https://github.com/dalek7/DLWorkshop17Summer

Rectified Linear Unit (ReLU)



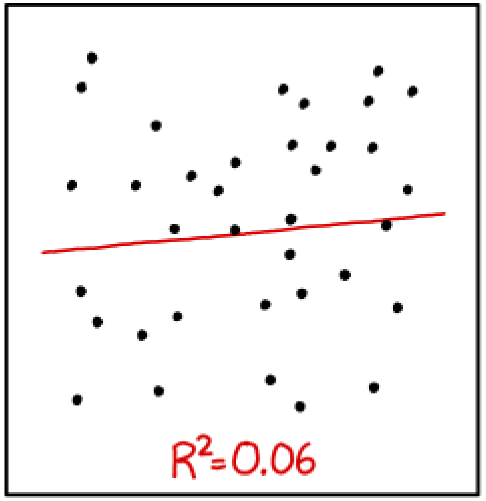


```
out = tf.nn.relu(y)
```

```
(-5, '-->',     )
(-4, '-->',     )
(-3, '-->',     )
(-2, '-->',     )
(-1, '-->',   ? )
( 0, '-->',     )
( 1, '-->',     )
( 2, '-->',     )
( 3, '-->',     )
( 4, '-->',     )
```

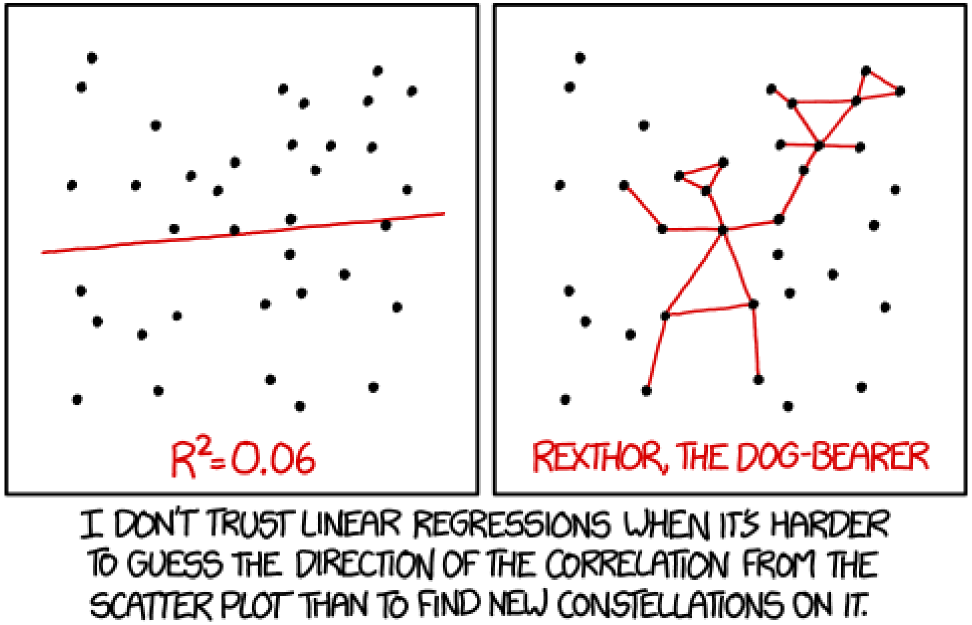
00-5-relutest.py 를 열어주세요

Linear Regression ??



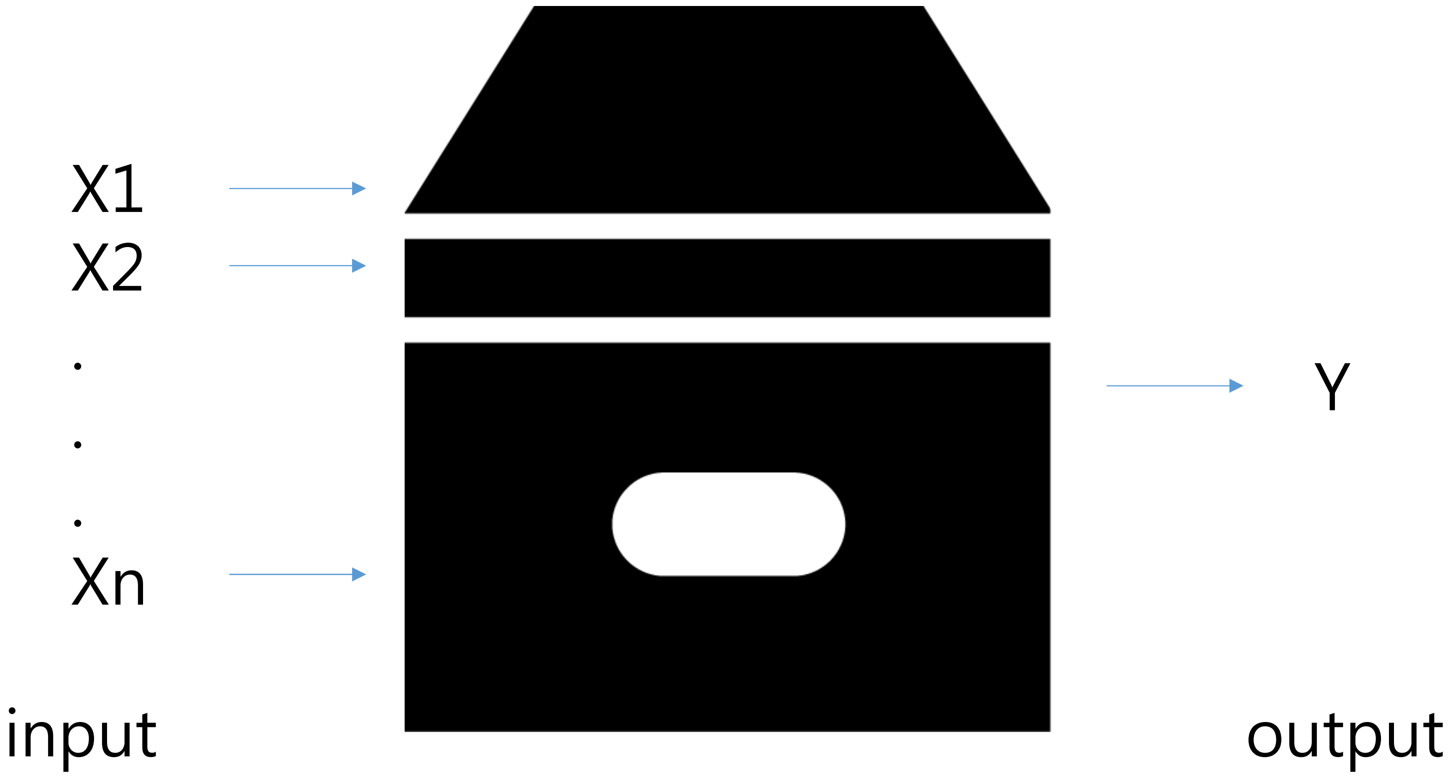
https://xkcd.com/1725/

Linear Regression vs Rexthor-the-dog-bearer regression


R²=0.06
REXTHOR, THE DOG-BEARER
I DON'T TRUST LINEAR REGRESSIONS WHEN IT'S HARDER TO GUESS THE DIRECTION OF THE CORRELATION FROM THE SCATTER PLOT THAN TO FIND NEW CONSTELLATIONS ON IT.

https://xkcd.com/1725/

Regression



X1

X2

.

.

.

Xn

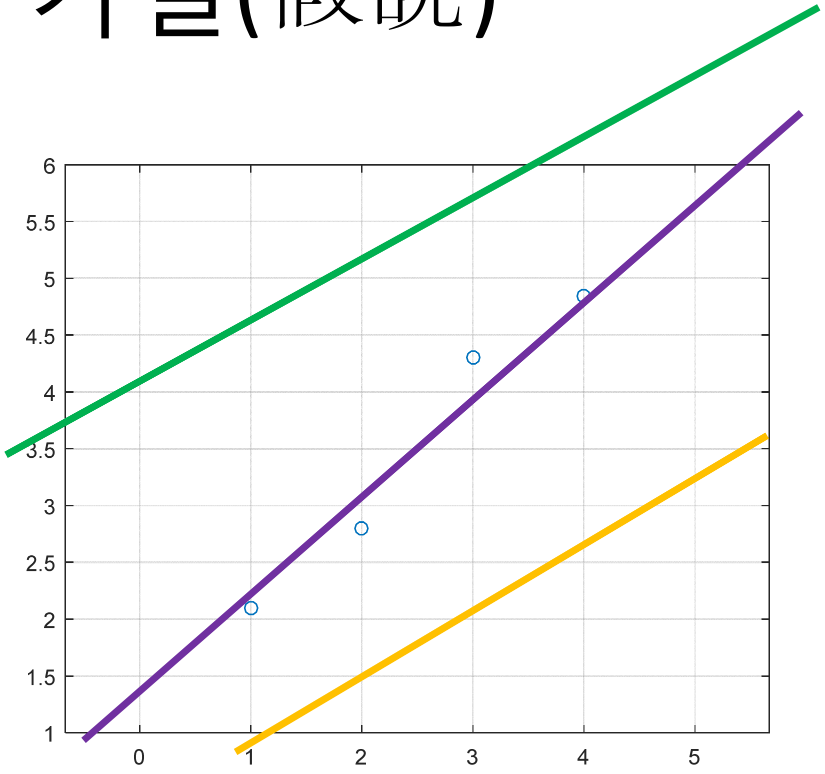
Y

input

output

(Linear) Hypothesis, 가설(假說)

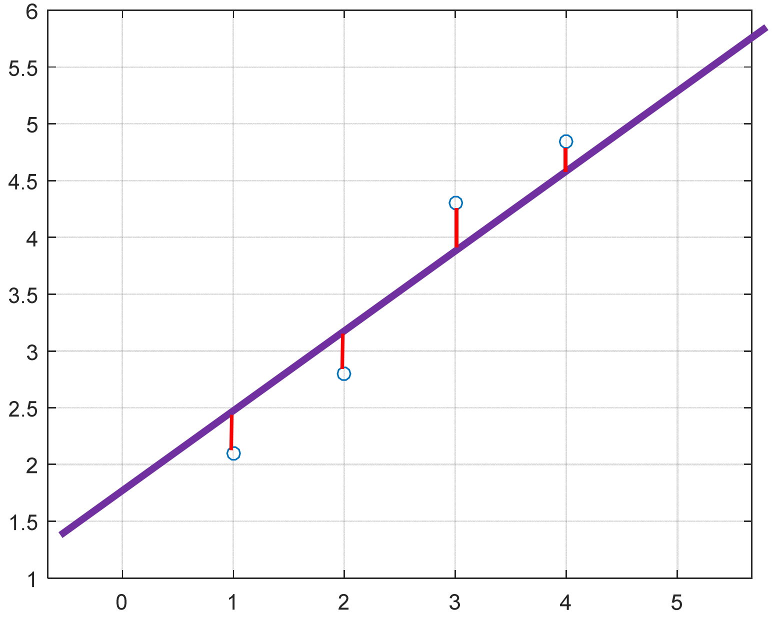
$$H(x) = Wx + b$$



https://www.slideshare.net/SRulez/linear-regression-machine-learning-term

Error ?

$$H(x) = Wx + b$$

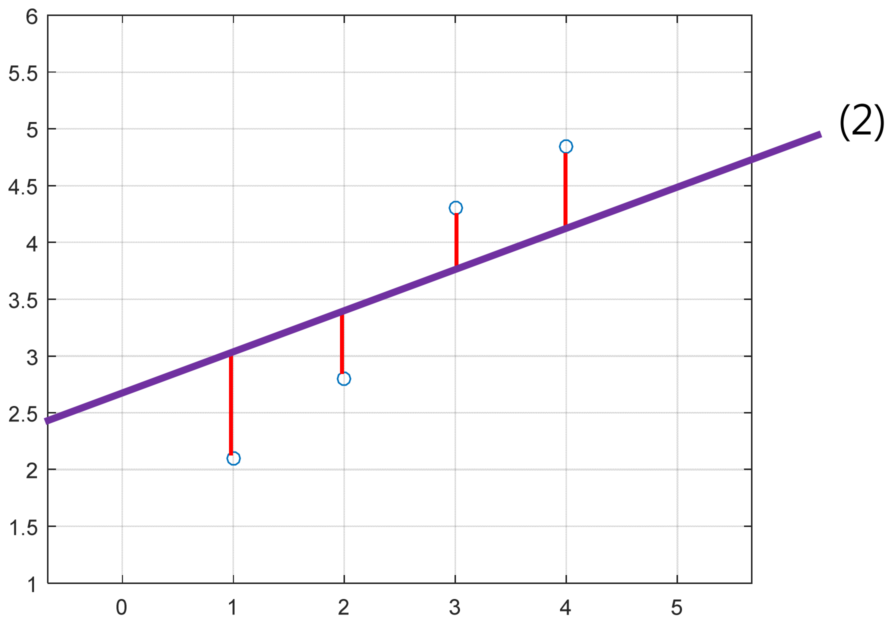


(1)

Error ?

W, b 에 따라서 에러 크기가 달라짐

$$H(x) = Wx + b$$



Linear Regression Hypothesis

$$H(x) = Wx$$

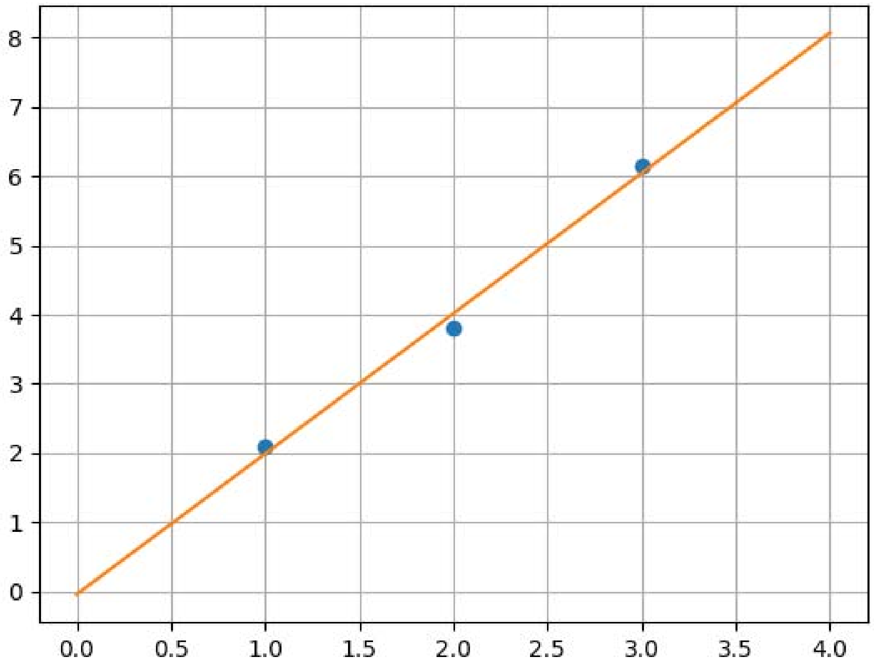
$$cost(W) = \frac{1}{m} \sum_{i=1}^{m} (Wx^{(i)} - y^{(i)})^2$$

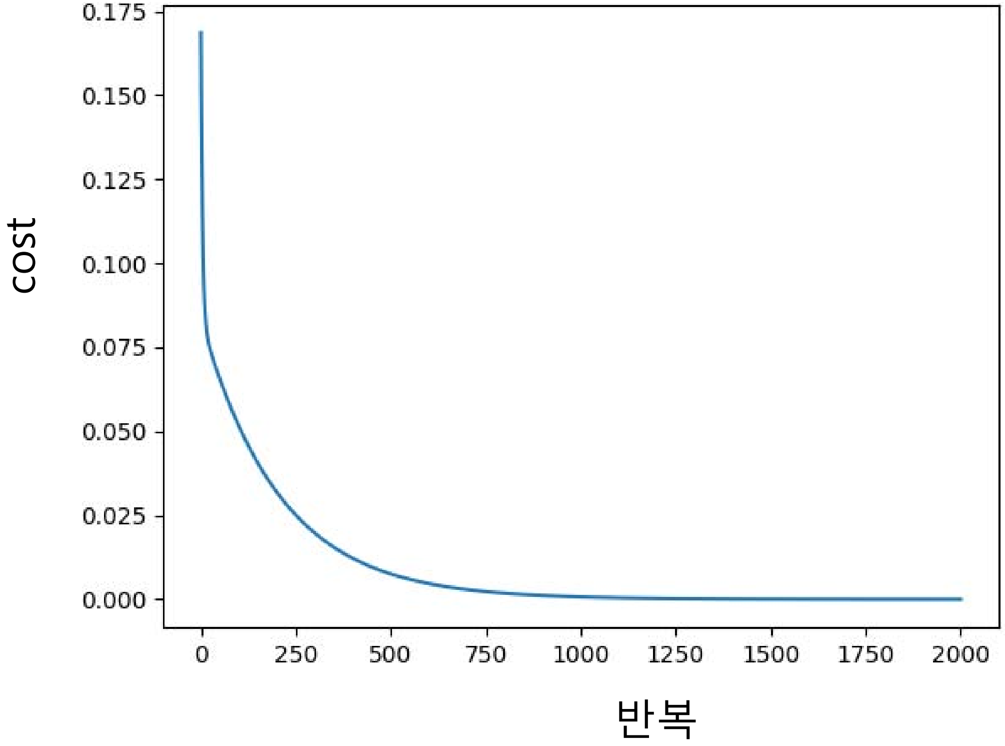
Cost function

- A function to be *minimized*
  최소화 되어져야할 값

$$\underset{W,b}{\text{minimize}}\, cost(W, b)$$

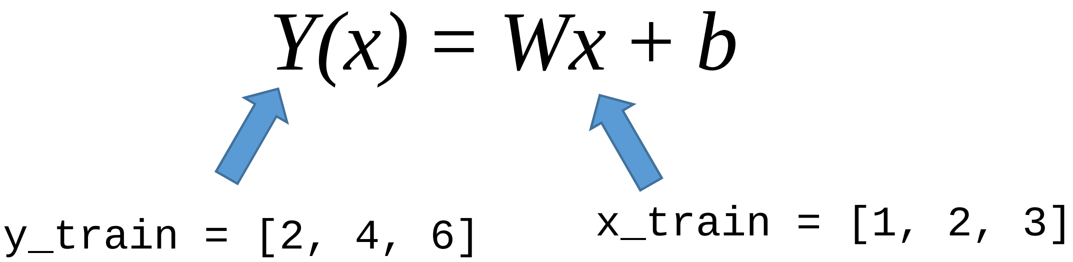
Linear regression 실습





01-1-linear_regression.py 를 열어주세요

Can you guess ?



$$Y(x) = Wx + b$$

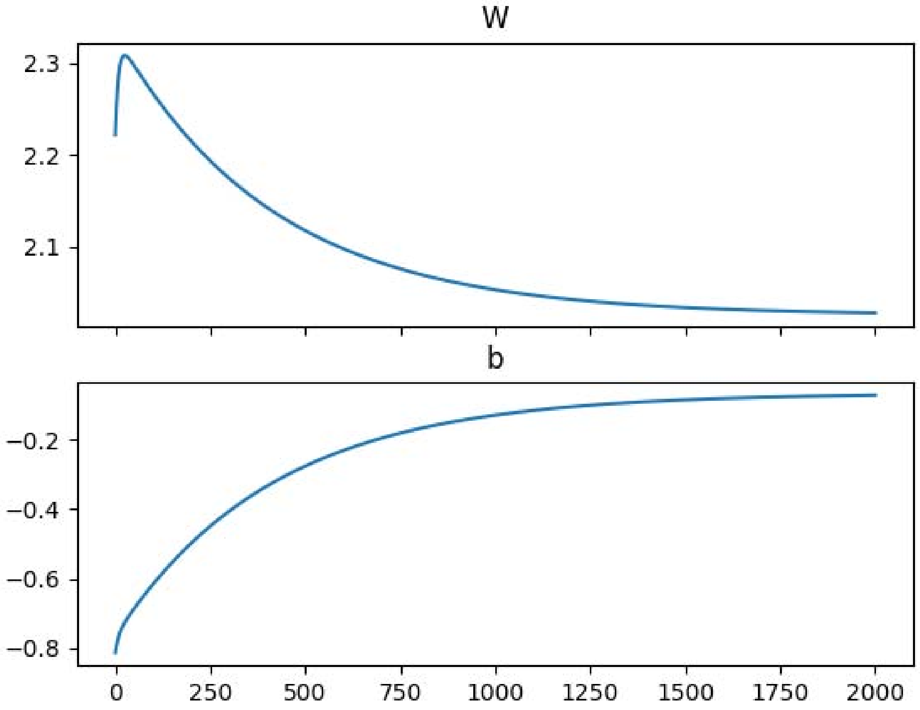
```
y_train = [2, 4, 6]
```

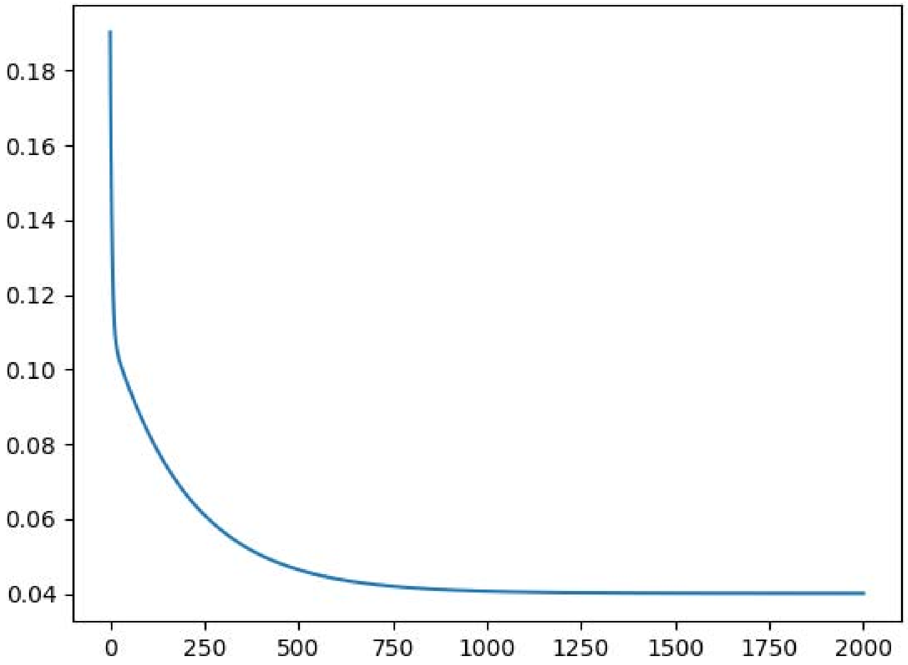
```
x_train = [1, 2, 3]
```

```
# X and Y data
x_train = [1, 2, 3]
y_train = [2, 4, 6]
```

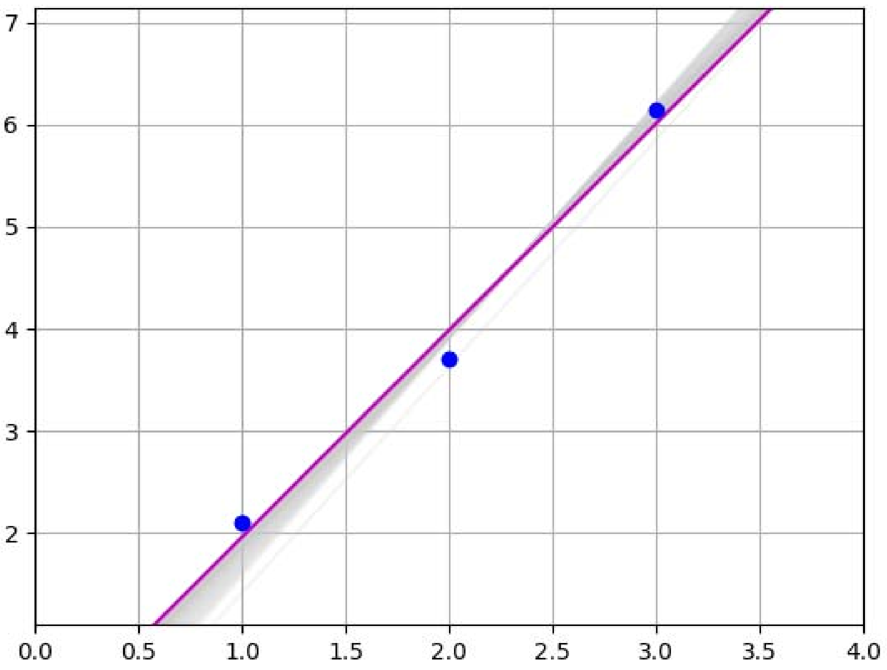
```
(0, 0.16853049, 2.2220104, -0.81235671)
(20, 0.076517023, 2.308851, -0.73138767)
(40, 0.068800062, 2.3035986, -0.69294065)
(60, 0.06247896, 2.2902124, -0.65998709)
(80, 0.056744356, 2.2766576, -0.62893307)
(100, 0.051536143, 2.2636638, -0.59937203)
(120, 0.04680597, 2.2512732, -0.57120341)
(140, 0.042509858, 2.2394645, -0.54435903)
(160, 0.038608193, 2.2282104, -0.5187763)

…

(1900, 8.896759e-06, 2.0034645, -0.0078751221)
(1920, 8.081005e-06, 2.0033016, -0.0075051254)
(1940, 7.3392598e-06, 2.0031466, -0.0071525616)
(1960, 6.6663715e-06, 2.0029988, -0.0068165772)
(1980, 6.0546249e-06, 2.0028582, -0.0064964173)
(2000, 5.498538e-06, 2.0027235, -0.0061912765)
```

반복과정 !


W
2.3
2.2
2.1
b
−0.2
−0.4
−0.6
−0.8
0
250
500
750
1000
1250
1500
1750
2000


0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0
250
500
750
1000
1250
1500
1750
2000

반복과정 !



01-1-linear_regression-optional.py

Testing our model !

- 학습된 가설/모델 검증해보기

01-1-linear-regression-placeholder.py
를 열어주세요

```
print("# Fit the line with new training data")
# Fit the line with new training data
for step in range(2001):
    cost_val, W_val, b_val, _ = \
        sess.run([cost, W, b, train],
                 feed_dict={X: [1, 2, 3, 4, 5],
                            Y: [2.1, 3.1, 4.1, 5.1, 6.1]})
    if step % 20 == 0:
        print(step, cost_val, W_val[0], b_val[0])

# Testing our model
print(sess.run(hypothesis, feed_dict={X: [5]}))
print(sess.run(hypothesis, feed_dict={X: [2.5]}))
print(sess.run(hypothesis, feed_dict={X: [1.5, 3.5]}))
```

```
[ 6.10045338]
[ 3.59963846]
[ 2.59931231  4.59996414]
```

오호라 !

- 다음 연립방정식의 해를 구하여라.

$$x + 2y = 6$$
$$x - 3y = 1$$

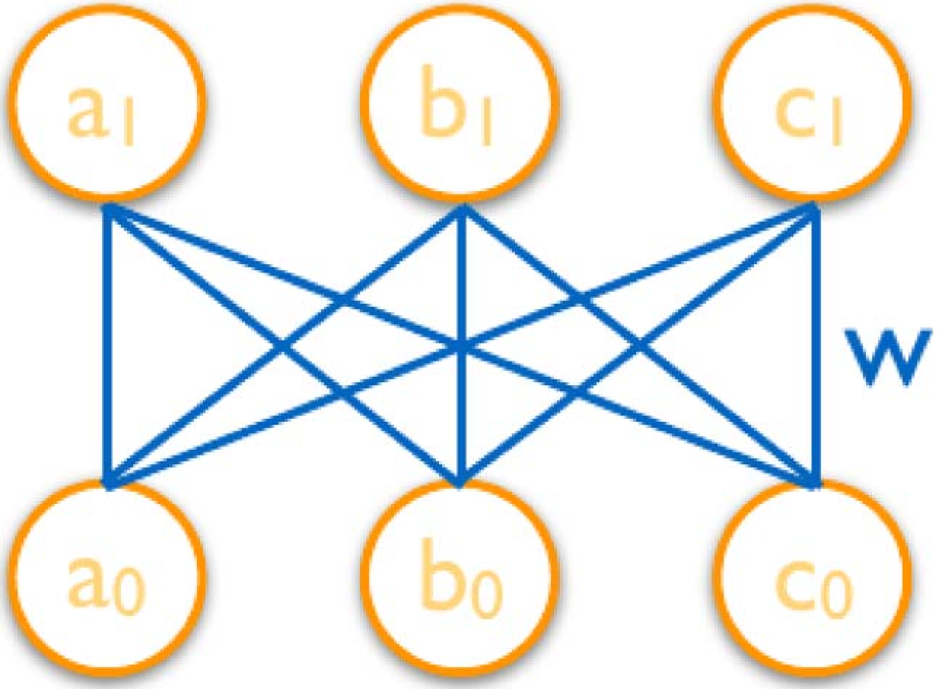
$$\begin{bmatrix} 1 & 2 \\ 1 & -3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 6 \\ 1 \end{bmatrix}$$

```
A = [[1.0, 2.0],
     [1.0, -3.0]]

b = [[6.0],
     [1.0]];
```

01-2-equation.py 를 열어주세요

http://mathbang.net/16

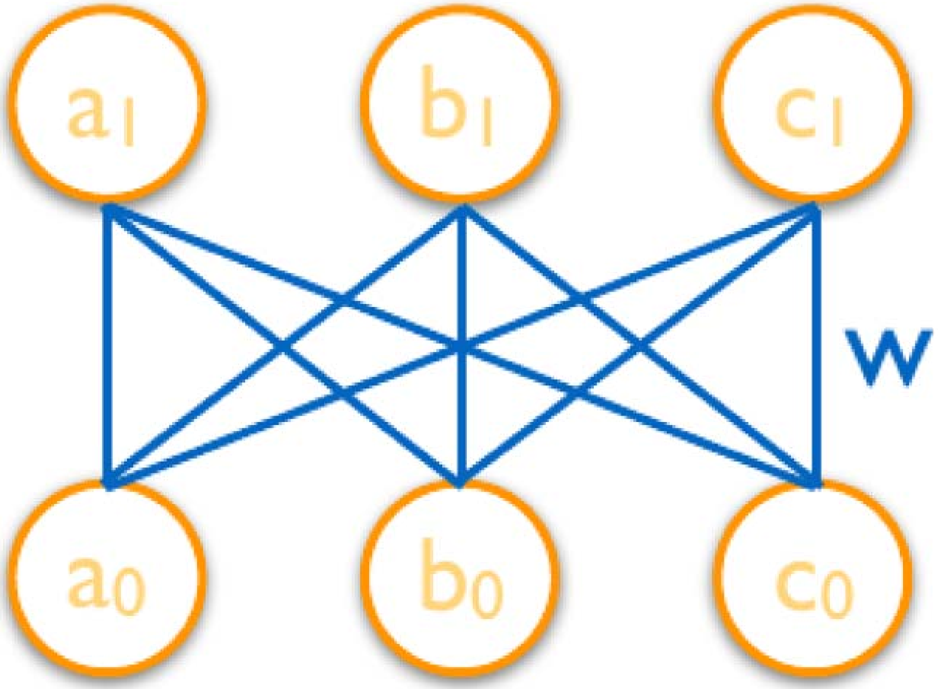
A simple Rectified Linear Unit (ReLU) network

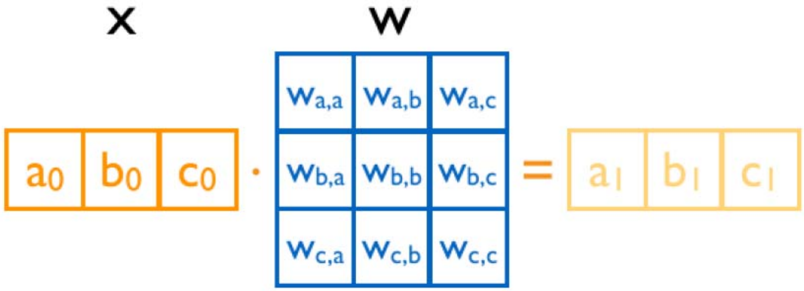


$$\begin{array}{c} x \\ \begin{bmatrix} a_0 & b_0 & c_0 \end{bmatrix} \end{array} \cdot \begin{array}{c} w \\ \begin{bmatrix} w_{a,a} & w_{a,b} & w_{a,c} \\ w_{b,a} & w_{b,b} & w_{b,c} \\ w_{c,a} & w_{c,b} & w_{c,c} \end{bmatrix} \end{array} = \begin{array}{c} y \\ \begin{bmatrix} a_1 & b_1 & c_1 \end{bmatrix} \end{array}$$

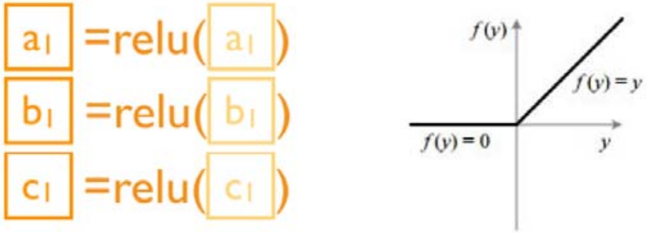
```
y = tf.matmul(x,w)
```

A simple Rectified Linear Unit (ReLU) network



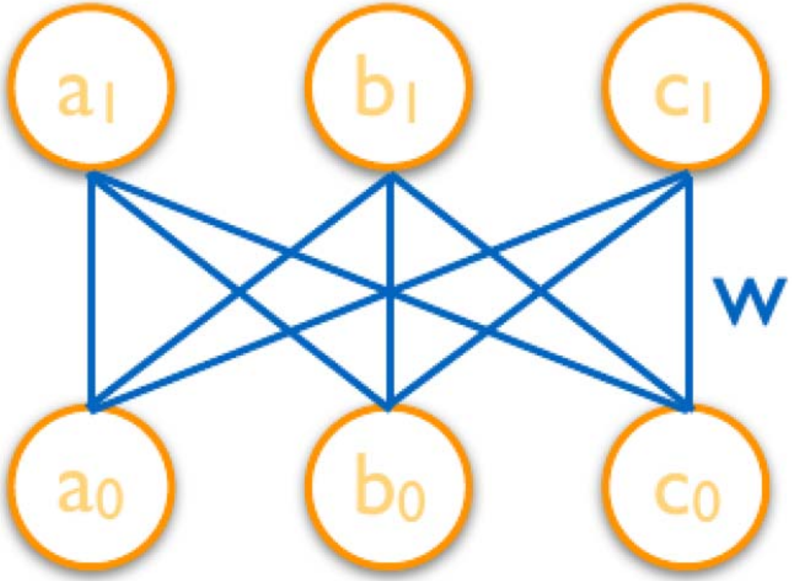


```
y = tf.matmul(x,w)
```



```
out = tf.nn.relu(y)
```

A simple Rectified Linear Unit (ReLU) network



```
import tensorflow as tf
sess = tf.Session()
x = tf.placeholder("float", [1, 3])
w = tf.Variable(tf.random_normal([3, 3]), name='w')
y = tf.matmul(x, w)
relu_out = tf.nn.relu(y)
```

01-3-simplenetwork.py 를 열어주세요

기타

- Matplotlib 연습해보기 https://matplotlib.org/examples/pylab_examples/subplot_demo.html

Acknowledgement

universitätbonn ais

모두를 위한 머신러닝/딥러닝 강의

Seung-chan

Seung-chan

Jeung-Chan

감사합니다.