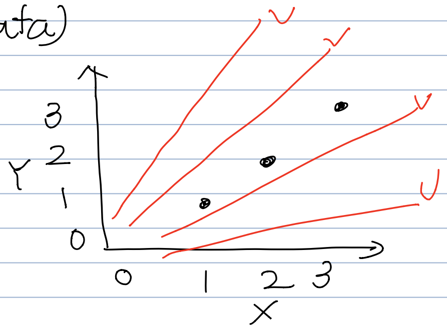
Section 2. Linear Regression의 개념

I. Linear Regression의 Hypothesis와 cost

1. Regression (data)

X	Y
1	1
2	2
3	3



선을 찾는 것이 학습하는 것임!!

가설 : (Linear) Hypothesis

$H(x) = Wx + b$

선과 점들과의 거리를 비교해서 점과 가까운 것을 고른다.

1) Cost function : 선과 점들과의 거리

$(H(x) - y)^2$ $H(x) = Wx + b$

$$Cost(W,b) = \frac{1}{m}\sum_{i=1}^{m}(H(x_i) - y_i)^2$$ m : 학습 data 개수

⇒ Cost가 가장 작은 W와 b를 구하는 것이 linear regression의 학습!

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

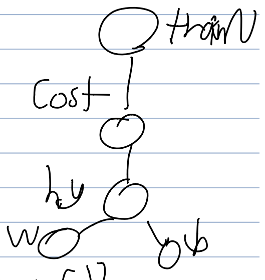
II. Tensorflow로 간단한 linear regression 구현

1. 구현 방법

1) Build graph

```
X_train = [1,2,3]
y_train = [1,2,3]
w = tf.Variable(tf.random_normal([1]), name='weight')
b = tf.Variable(tf.random_normal([1]), name='bias')
hypothesis = x_train * W + b
```

```
cost = tf.reduce_mean(tf.square(hypothesis - y_train))
t = [1., 2., 3., 4.]
tf.reduce_mean(t) #평균구함
optimizer = tf.train.GradientDescentOptimizer
                (learning_rate = 0.01)
train = optimizer.minimize(cost)
```



2) Run/Update graph

```
sess = tf.Session()
sess.run(tf.global_variables_initializer())
for step in range(2001):
    sess.run(train)
    if step % 20 == 0:
        print(step, sess.run(cost), sess.run(w), sess.run(b))
```

필요할 때 값 넣어줌

2. Placeholder를 이용하여 구현

```
X = tf.placeholder(tf.float32)
Y = tf.placeholder(tf.float32)

for step in range(2001):
    cost_val, W_val, b_val, _ = \
        sess.run([cost, W, b, train],
                 feed_dict={X:[1,2,3], Y=[1,2,3]})

    if step % 20 == 0:
        print(step, cost_val, W_val, b_val)
```

3. 학습이 잘 되었나 확인

```
print(sess.run(hypothesis, feed_dict={X:[?]}))
```