## ML Lec 01 - 기본적인 Machine Learning의 용어와 개념 설명

### ML

- due to limitation of explicit programming(spam filter,automatic driving has too many rules)

- field of study that gives computers the ability to learn without being explicitly programmed (Arthur Samuel(1959)

### 1) Supervised learning

learning with labeled examples - training set



Most common problem type in ML

* Image labeling, Email spam filter, Predicting exam score
* from previous score(1~100) based on time spent -> **regression**
* pass/n-pass -> **binary classification**
* Letter grade(A,B,C,E and F) based on time spent : **multi-label classification**

with Training data set

### 2) Unsupervised learning: un-labeled data

ex) google news grouping / word clustering

## 

## 

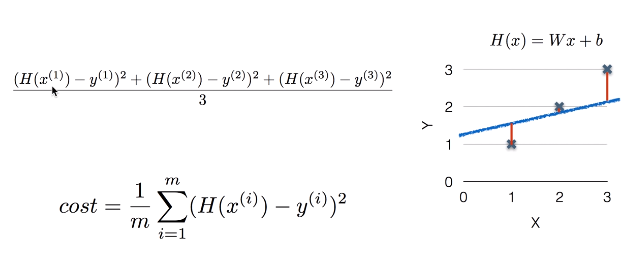
## 02) Linear Regression의 Hypothesis와 cost

### regression: ex) predicting exam score

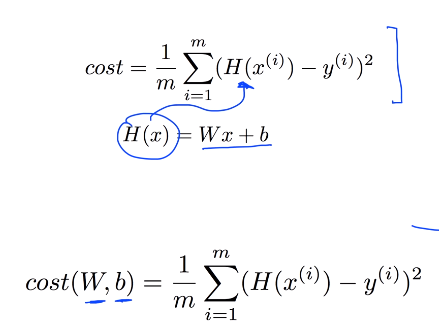
### Hypothesis: H(x) = Wx+b

### Cost function = Loss function

모델이 얼마나 실제에 가까운가? How fit the line to our data?



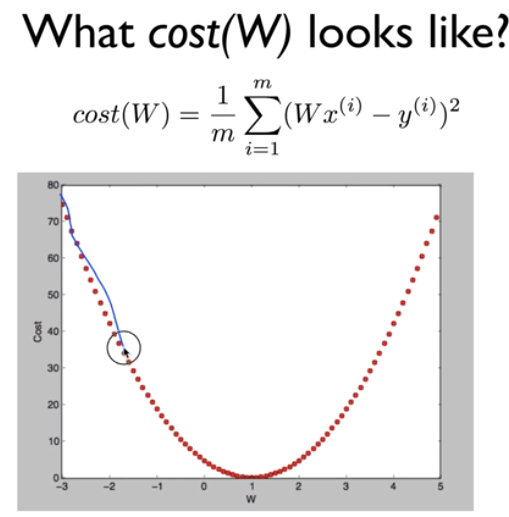
cost function을 아래와 같이 정리할 수 있는데, 학습을 한다는 것은 minimize cost(W,b)한 W와 b를 찾는 것이다.



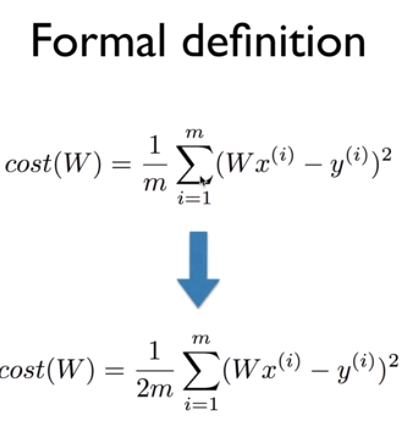
## 03) Linear Regressioin의 최소화 알고리즘의 원리 설명

### Gradient descent algorithm

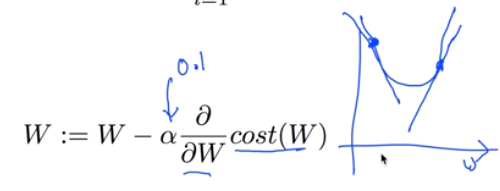
cost function이 최소화되는 점은 기울기가 최소화된 지점이다.



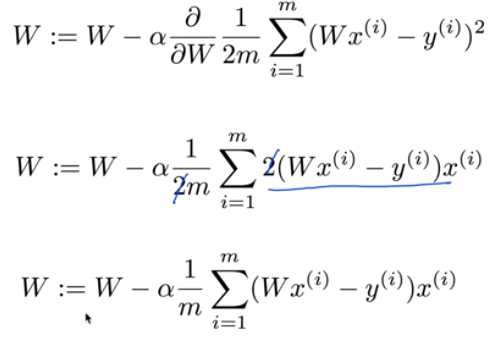
cost function을 미분한다.



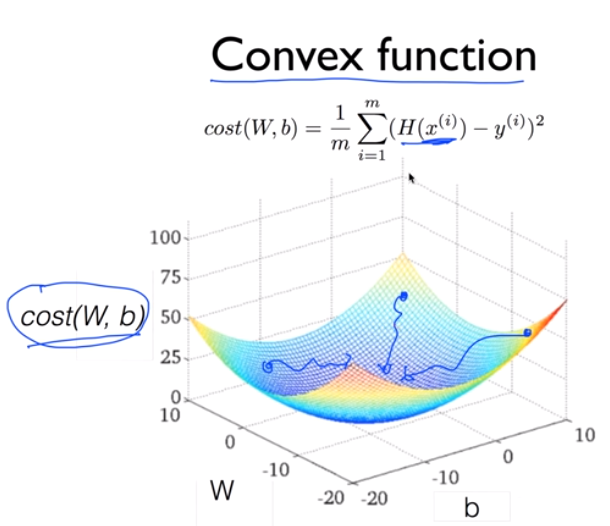
cost(w)를 미분했을 때 양/음 값이 나오면 w값을 -/+조절해본다.



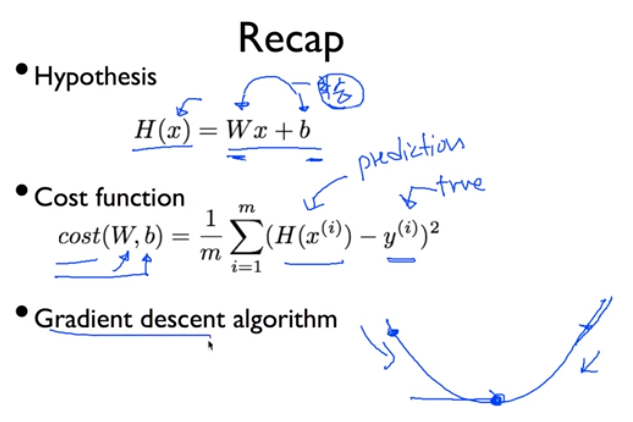
미분식 풀이



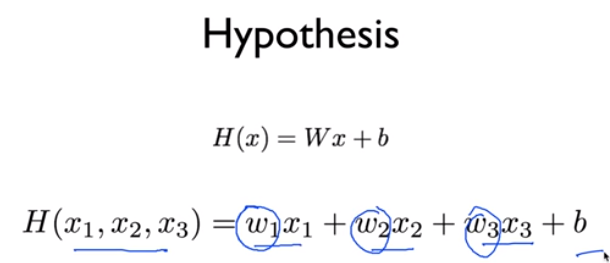
이때 주의: cost(W,b)가 convex function모양을 하고 있어야 성립됨



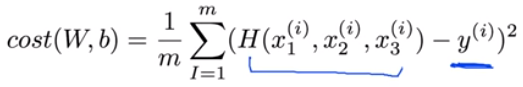
## 04) multi-variable linear regression



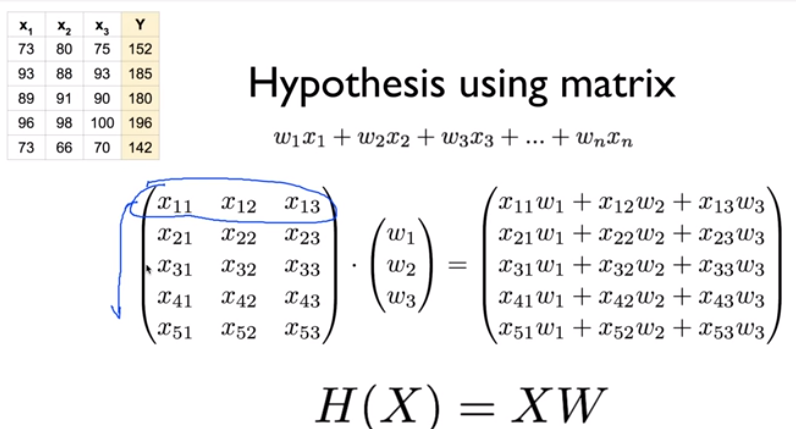
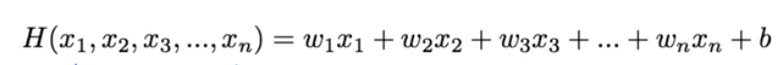
### regression using three inputs(x1,x2,x3)의 Hypothesis



### Cost function



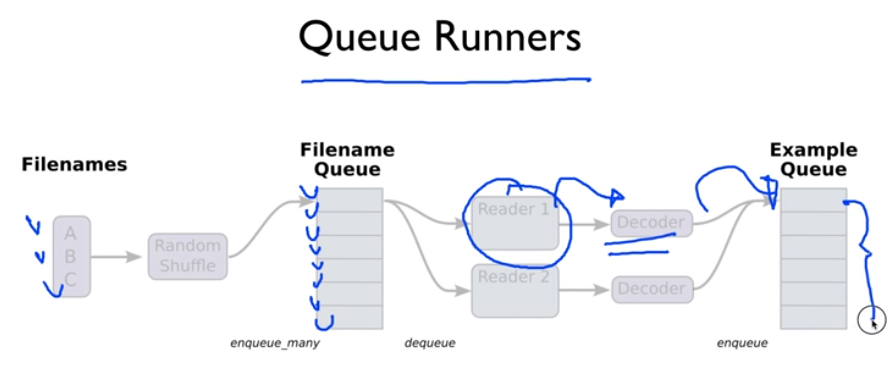
### x의 갯수가 증가한다면? -> Matrix



X의 [a,b] -> a: instance 수 / b: variable 수

H(x)의 [a,b] -> a: instance 수 / b: y값

### Queue Runners



## 05-1) Logistic Classification의 가설 함수 정의

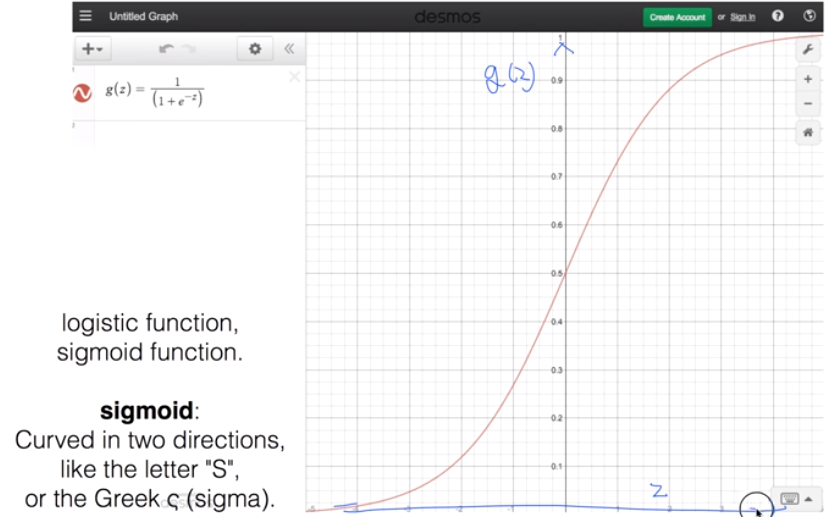
### Binary Classification

0,1 encoding으로 나타낸다

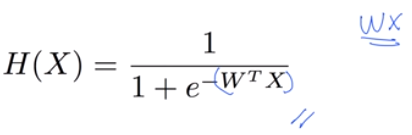
spam(1) or ham(0) / facebook: show(1) or hide(0) / …

기존 Linear regression의 hypothesis에서는 값이 0이하 or 1이상의 수가 나온다.

but binary classification에서는 0 ~ 1만 나와야 한다. => **sigmoid = logistic function**



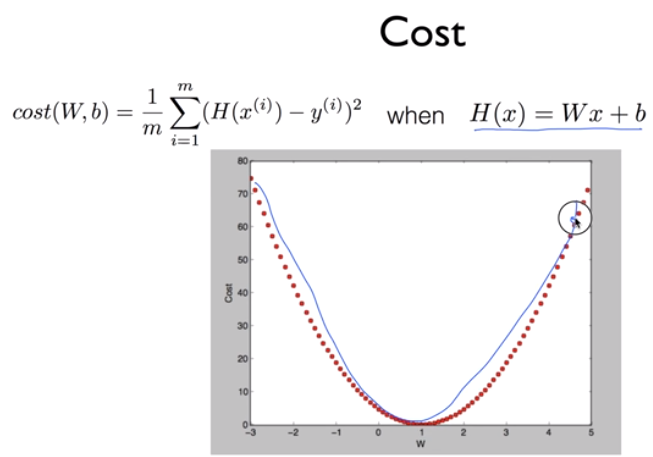
### Logistic Hypothesis



## 

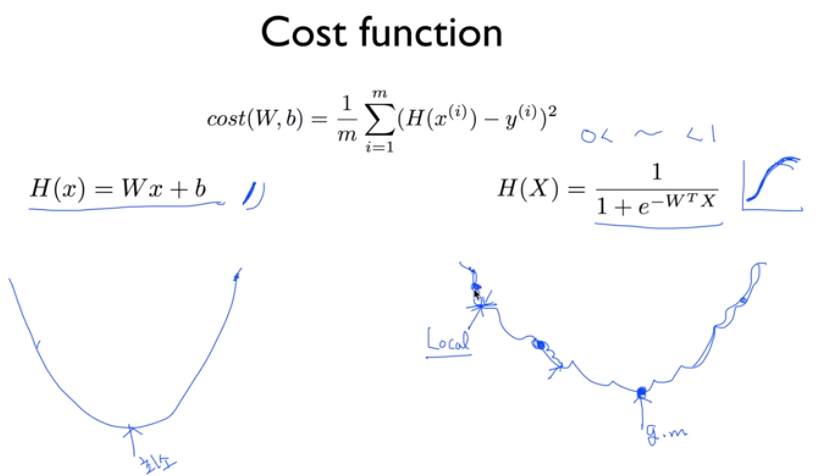
## 05-2) Logistic Regression의 cost 함수 설명

### 기존 linear regression의 cost function



### 기존 Logistic regression cost function의 문제점

기존 linear regression의 cost function은 이차방정식처럼 기울기=0인 지점이 하나이지만,



새로운 logistic hypothesis의 cost function을 그리게 되면 울퉁불퉁해진다.

우리의 목표인 global minimum이 아닌, local minimum에서 멈추게 되므로 새로운 cost function 식이 필요하다.

### 새로운 cost function 이론

classification에서 y의 값은 0과 1밖에 없으므로 y=0일때, y=1일때로 나누어서 생각한다.

1) y=0일때,

H(x): 0이라면, cost값은 0가 나와야 한다.

H(x): 1이라면, cost값은 무한대로 커질 수밖에 없다.

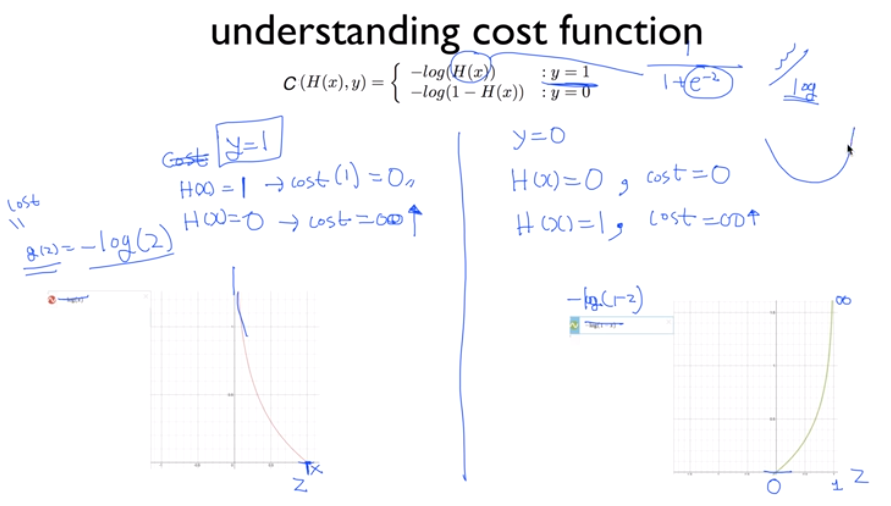
2) y=1일때,

H(x): 1이라면, cost = 0

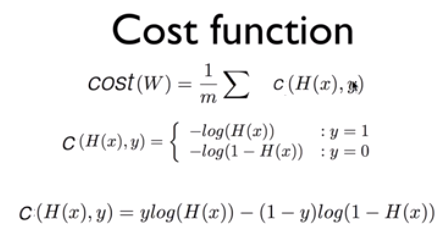
H(x): 0이라면, cost는 무한대로 커진다.

이를 정리하면 아래와 같은 그림과 식이 나온다.

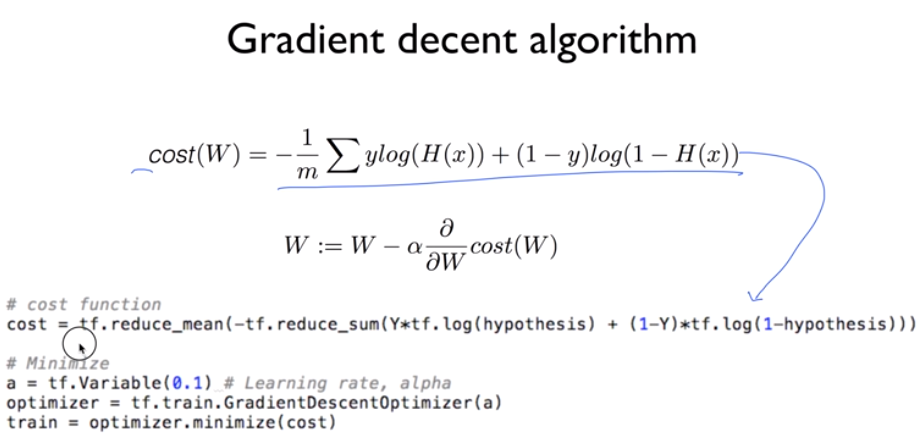
아래 두 그래프를 합치면 linear regression의 cost function함수 모양(아래볼록)이 나옴



### cost function



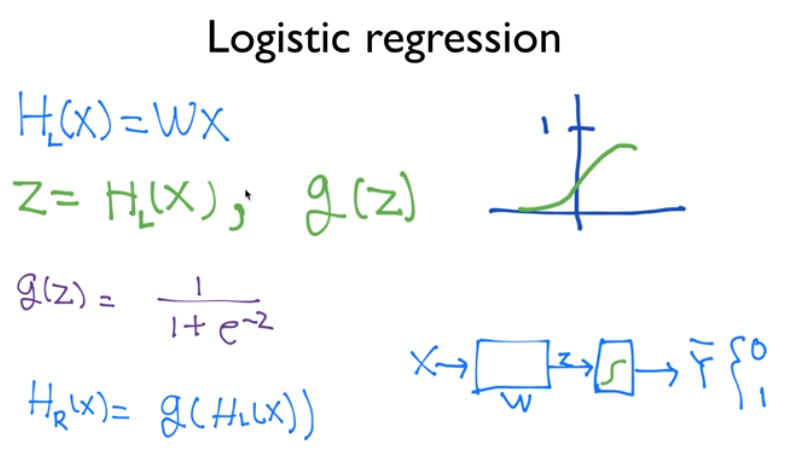
### Minimize cost - Gradient descent algorithm



## 06-1) Softmax Regression

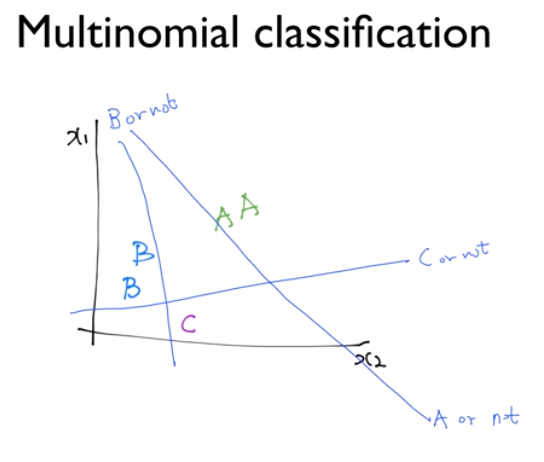
multinomial classification 中 softmax classification

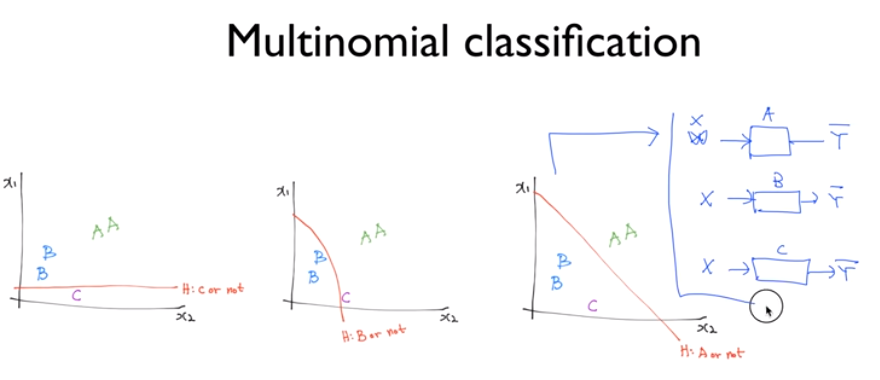
기존 logistic regressioin은 w기울기를 가진 x를 넣어서 z를 가지는데, 결과값을 0~1사이로 만들기 위해서 z를 다시 sigmoid함수에 넣는다. 이 값을 보통 y의 hat이라고 한다.



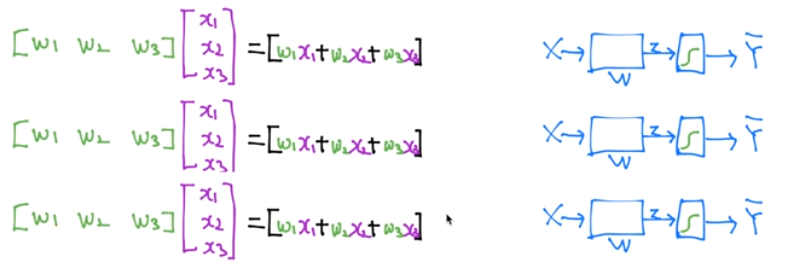
### Multinomial classification

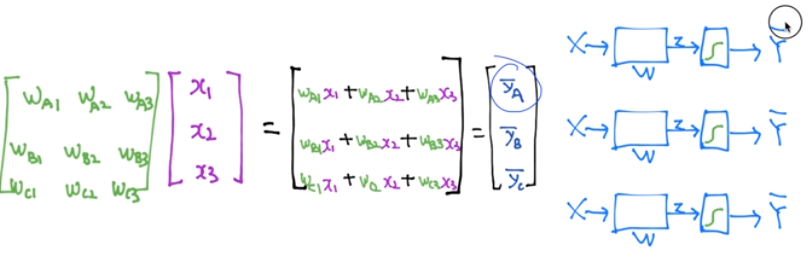
N개의 classification을 갖는다면, N개의 binary classification으로 나타낼 수 있다.





이것을 식으로 나타내면 아래와 같고, 이를 묶어서 다시 나타내면 그 아래와 같다.





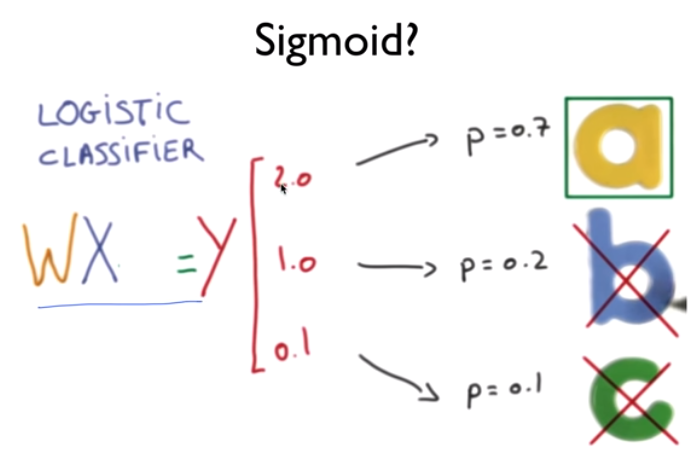
### Softmax

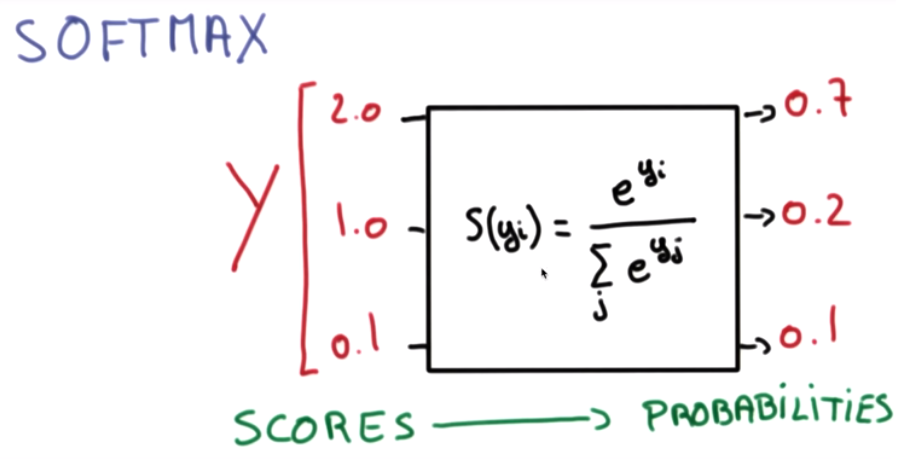
각 classification의 y hat 값들이 0~1 사이에 나올 수 있도록 sigmoid함수에 넣어준다.

이때 각 classification의 sigmoid값 합은 1이 될 수 있도록 만들 수 있다. like 확률

이것을 softmax함수라고 한다.

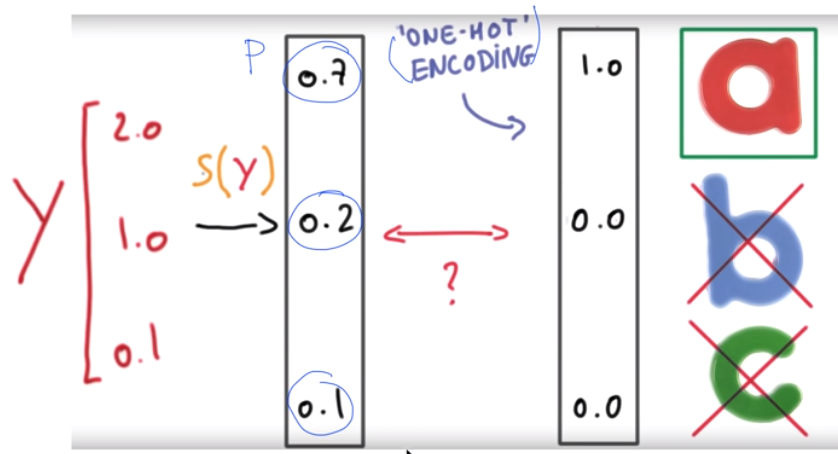
**특징1) 0~1 사이의 값을 갖는다. / 특징2) 전체의 합은 1이다.**





softmax함수로 나온 값을 0 or 1로 변환시켜 classificatioin을 구분하는 코딩을

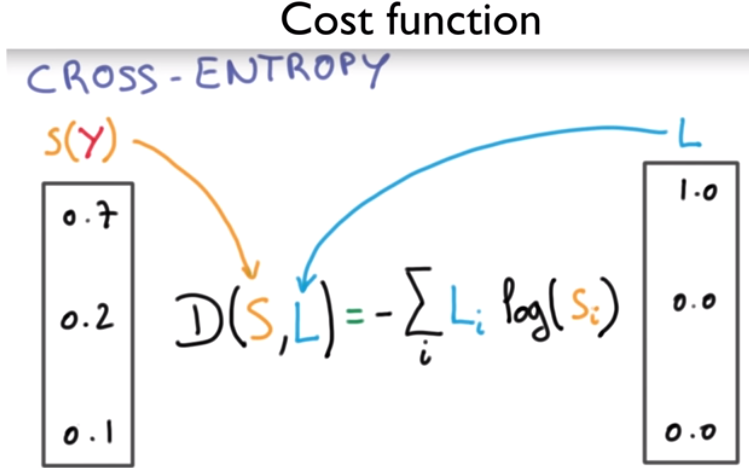
**one-hot encoding**이라고 한다.



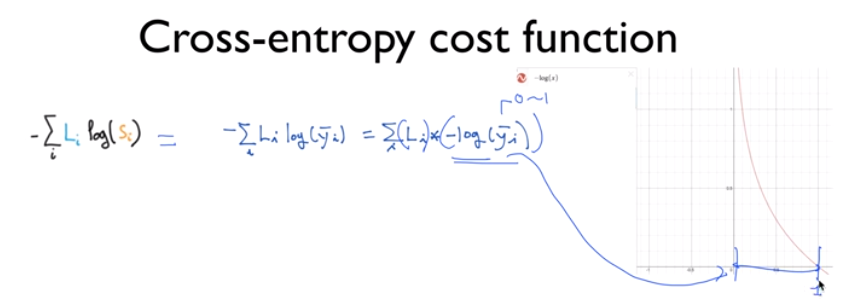
### cost function - cross entropy

softmax 함수를 넣어 예측한 값, s(y) = y hat 과 실제 값인 L(label)이 있다.

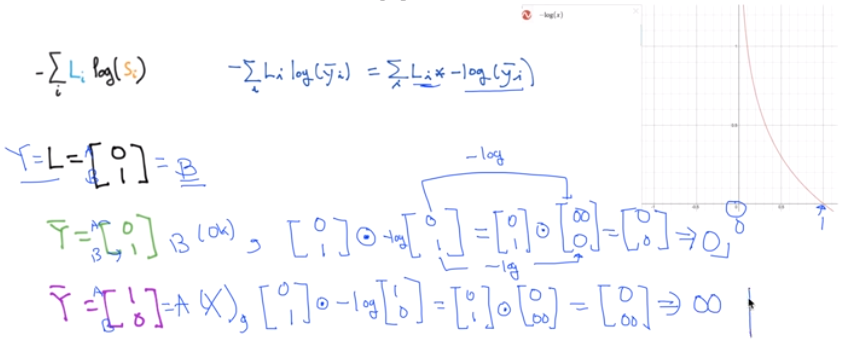
cross-entropy를 활용하여 둘 사이의 차이에 대한 cost function을 구해준다.



cross-entropy 식을 재정렬하면 아래와 같다. 이때 -log yi의 함수는 오른쪽 모양이다.

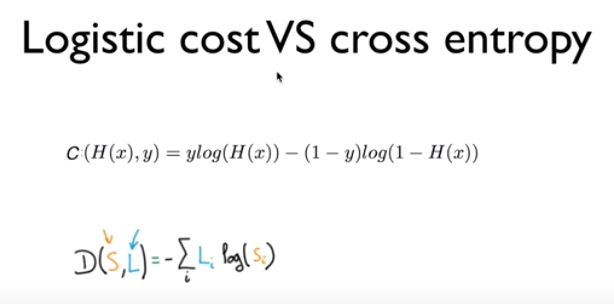


L(실제값)이 B이고, Y(예측값)이 각각 B와 A일 때, cost function은 0과 무한대가 나온다는 것을 알 수 있다.



### Logistic cost vs cross entropy

사실 두 cost function은 같다.

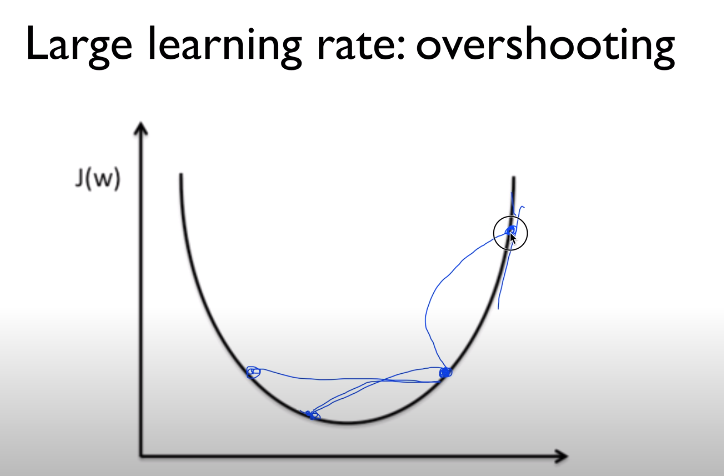


## 07-1) learning rate, overfitting, 그리고 일반화(regularization)

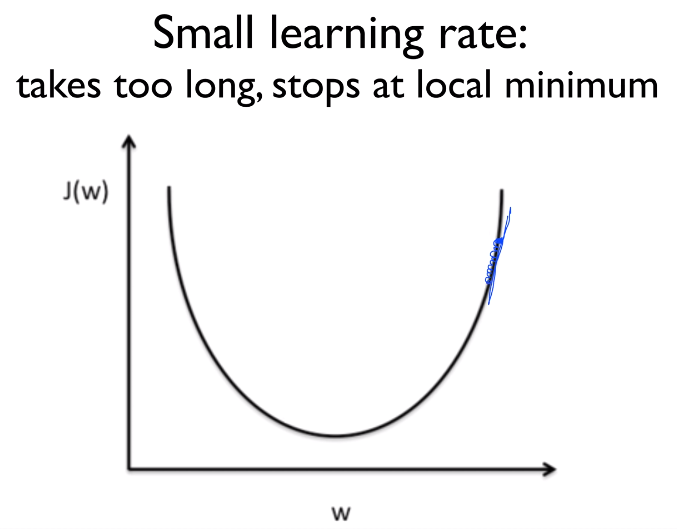
### learning rate

0.01을 기준으로 여러 값을 넣어 cost function을 test해야 한다.

Learning rate가 클 경우, overshooting될 수 있다.

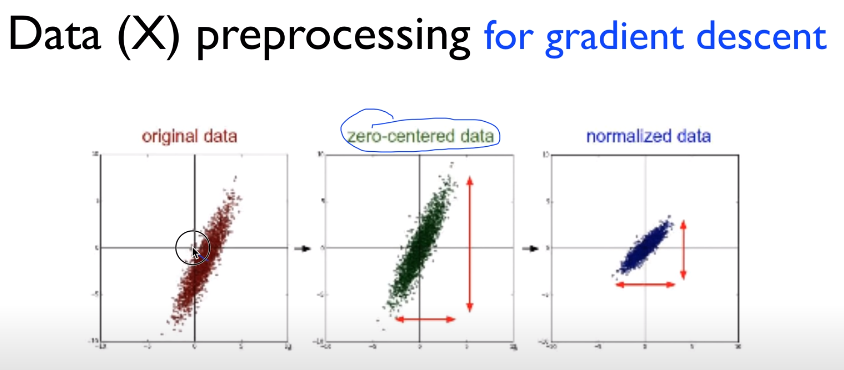


Learning rate가 작을 경우, 시간이 오래걸려 stop될 수 있다.



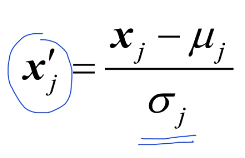
### Data(x) preprocessing for gradient descent

data x1과 x2의 범위가 확연히 다를 때, zero-centered data 혹은 normalized data 처리를 해야함



normalized data는 standardization을 통해 가능하다.

파이썬>> x\_std[:,0] = (x[:,0] - x[:,0].mean()) / x[:,0].std()



### overfitting

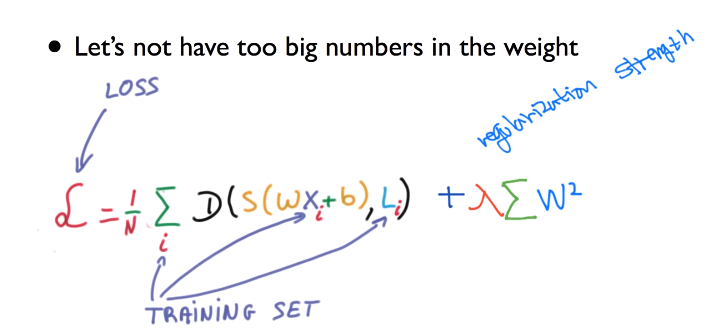
학습데이터에만 잘 학습되어서 실제 데이터에는 안 맞는 경우

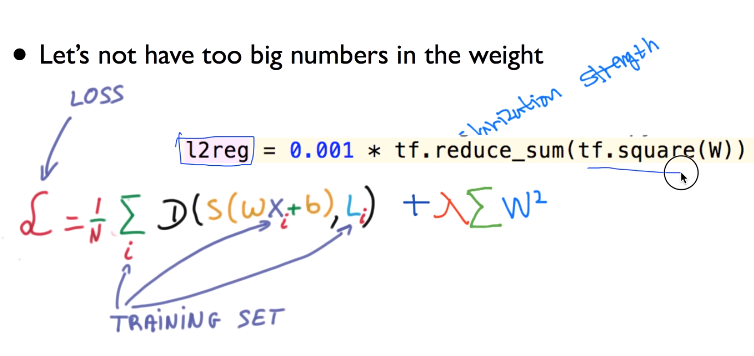
해결방법) more trainig data / reduce the number of featues / regularization

**regularization**

weight이 큰 값을 가지지 말자.

cost 함수 뒤에 텀 추가





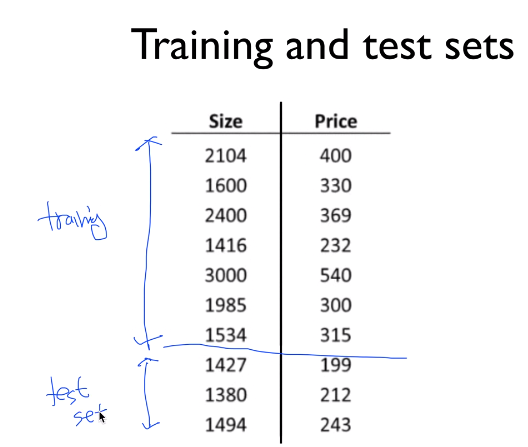
## 

## 07-2) Training/Testing Data set

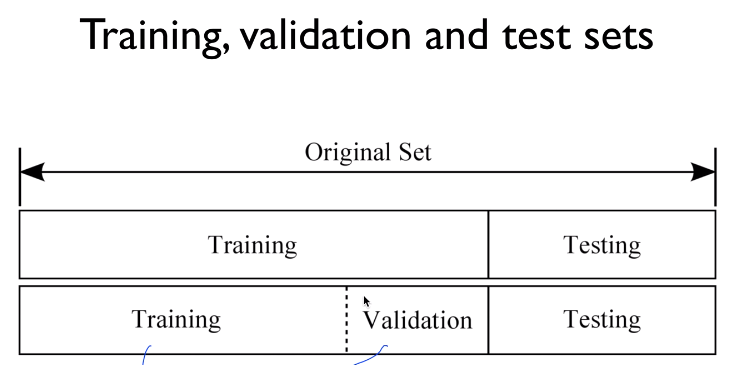
머신러닝 모델 평가하기

all of training set으로 평가하는 것 X

training set 과 test set으로 나누어 training set으로 학습 후 test set으로 평가



나아가 learning rate와 standardization의 텀 난다 값으로 데이터를 튜닝하기 위한 validation도 구분해주면 좋음.



### online learning

100만개의 data가 있으면 10만개씩 잘라서 순차적으로 학습시키며 모델에 추가하는 방법

### Accuracy

예측값과 실제값을 비교해서 95~99%가 나오면 good

08-1) 딥러닝의 기본 개념: 시작과 XOR 문제