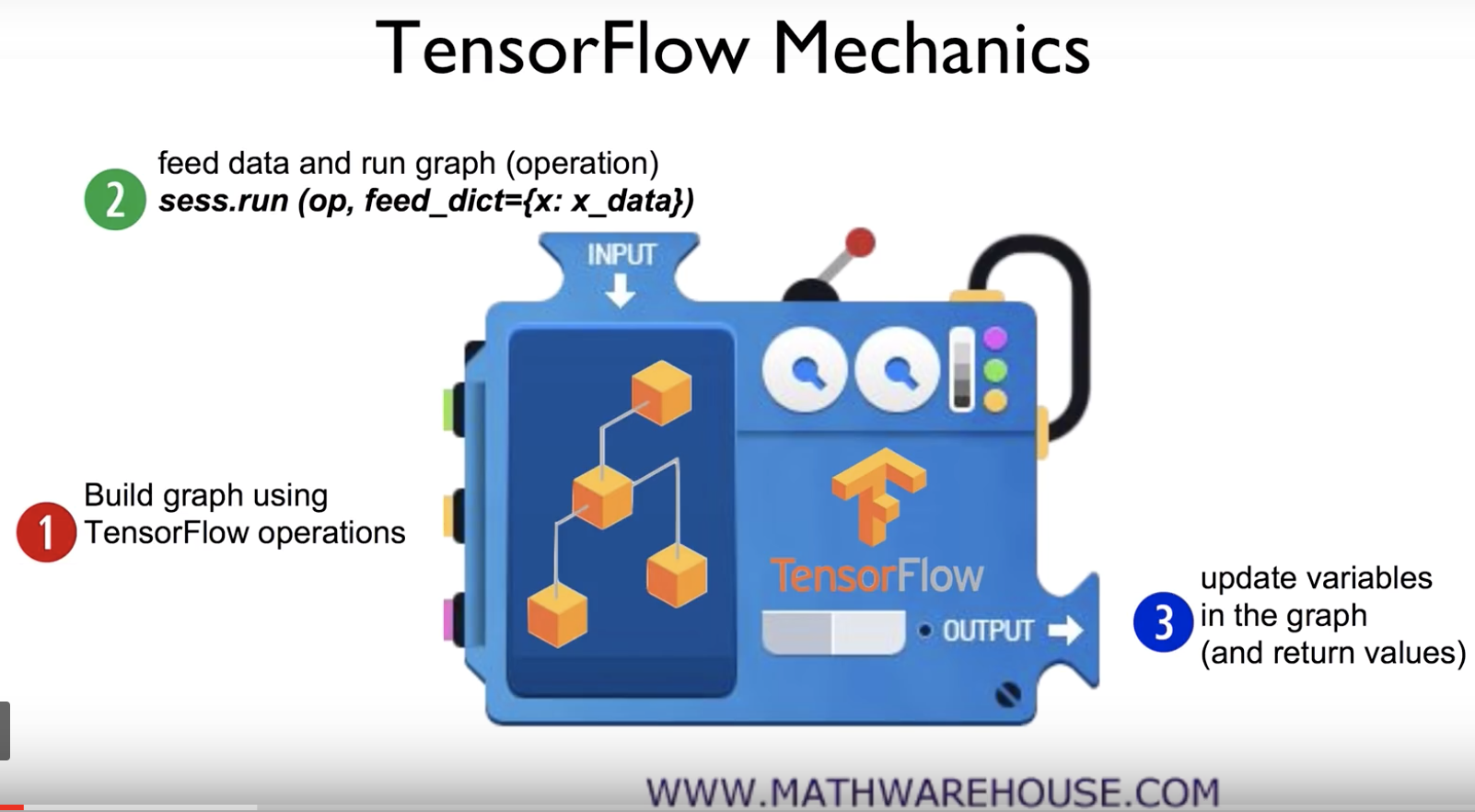
**섹션 2. Linear Regression의 개념**

**Linear Regression의 Hypothesis와 cost**

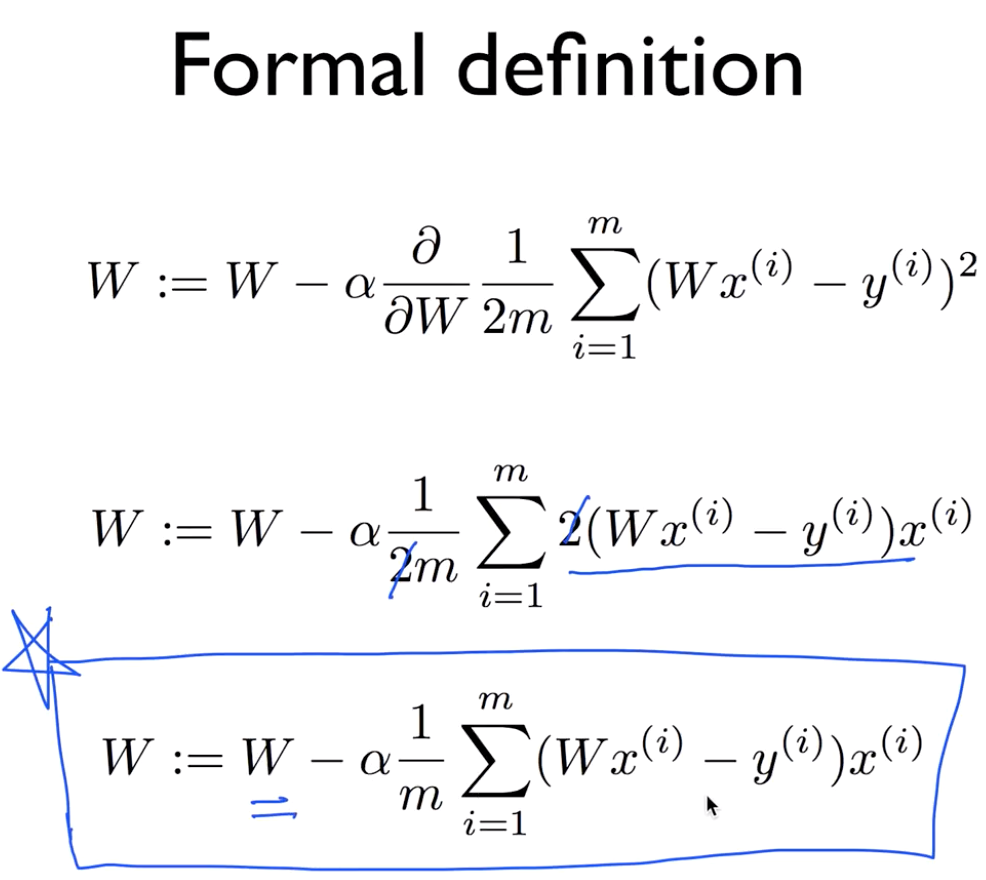
1. Linear Regression
   1. x를 넣었을 때 y를 얻기
   2. Supervised Learning
      1. (x, y) 쌍을 몇 개 주고, 새로운 x에 대한 y값을 얻기
2. Hypothesis
   1. Regression model의 학습을 위한 가정
   2. Linear Hypothesis
      1. Linear한 모델이 우리가 갖고 있는 데이터에 맞을 것이라고 가설을 세움
      2. 세상에는 linear model로 설명할 수 있는 경우가 많음
         1. 집이 클수록 집값이 높아짐학습을 하는 것은 이차원 그래프에서 데이터에 맞는 linear한 선을 찾는 것과 같음
3. Cost
   1. 어떤 선이 우리의 데이터에 가장 fit 하는가?(Which hypothesis is better?)
      1. H(x) = Wx + b
         1. 가설과 점의 거리 비교
         2. 가까울 수록 좋음
   2. cost function(loss function)
      1. H(x) - y: 양수가 될 수도, 음수가 될 수도 있으므로 좋지 않은 모델임
      2. **(H(x) - y)^2**
         1. **cost function**
         2. 양수
         3. 차이가 클수록 값이 더 커짐
   3. **cost = 1/m \* (sum of (H(x(i)-y(i))^2) (for i=1 to m)**
      1. goal: minimize cost(W, b)
      2. **cost를 가장 작게 하는 W와 b를 구하는 것이 학습의 목표**

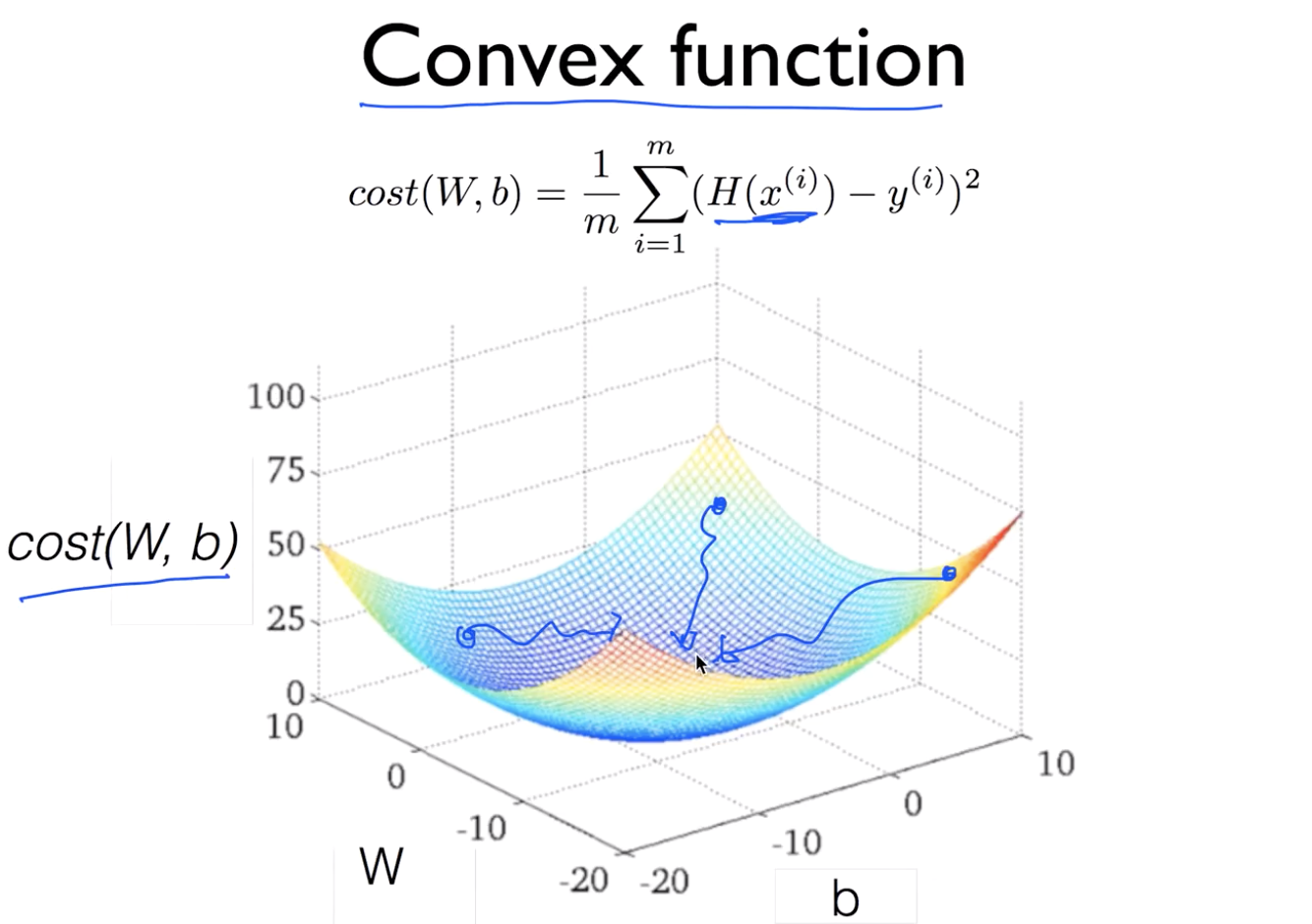


**섹션 3. Linear Regression cost 함수 최소화**

**Linear Regression의 cost 최소화 알고리즘의 원리**

1. Simplified hypothesis
   1. H(x) = Wx
   2. cost(W) = 1/m \* sum of (Wx(i) - y(i))^2 (for i=1 to m)
   3. W=1, cost(W)=?
      1. cost(W)=0
      2. W=0, cost(W)=4.67
      3. W=2, cost(W)=4.67
2. Gradient descent algorithm
   1. 경사를 따라 내려가는 알고리즘
   2. to minimize cost function
      1. 주어진 cost function cost(W, b)에 대해 cost를 최소화하는 W, b를 찾아줌
   3. can be applied to more general function: cost(w1, w2, …)
   4. 원리
      1. 0, 0 또는 다른 value에서 시작
      2. W, b를 조금씩 바꿔가면서 try & reduce cost(W, b)
      3. parameter를 바꿀 때마다 cost(W, b)를 가장 많이 줄일 수 있는 gradient를 선택(경사를 따라 한발자국 움직이고 … )
      4. local minimum에 수렴할 때까지 반복
      5. interesting property
         1. where you start can determine which minimum you end up
         2. 어떤 점에서 시작해도 최소점에 도달할 수 있다.



gradient descent algorithm을 사용하려면

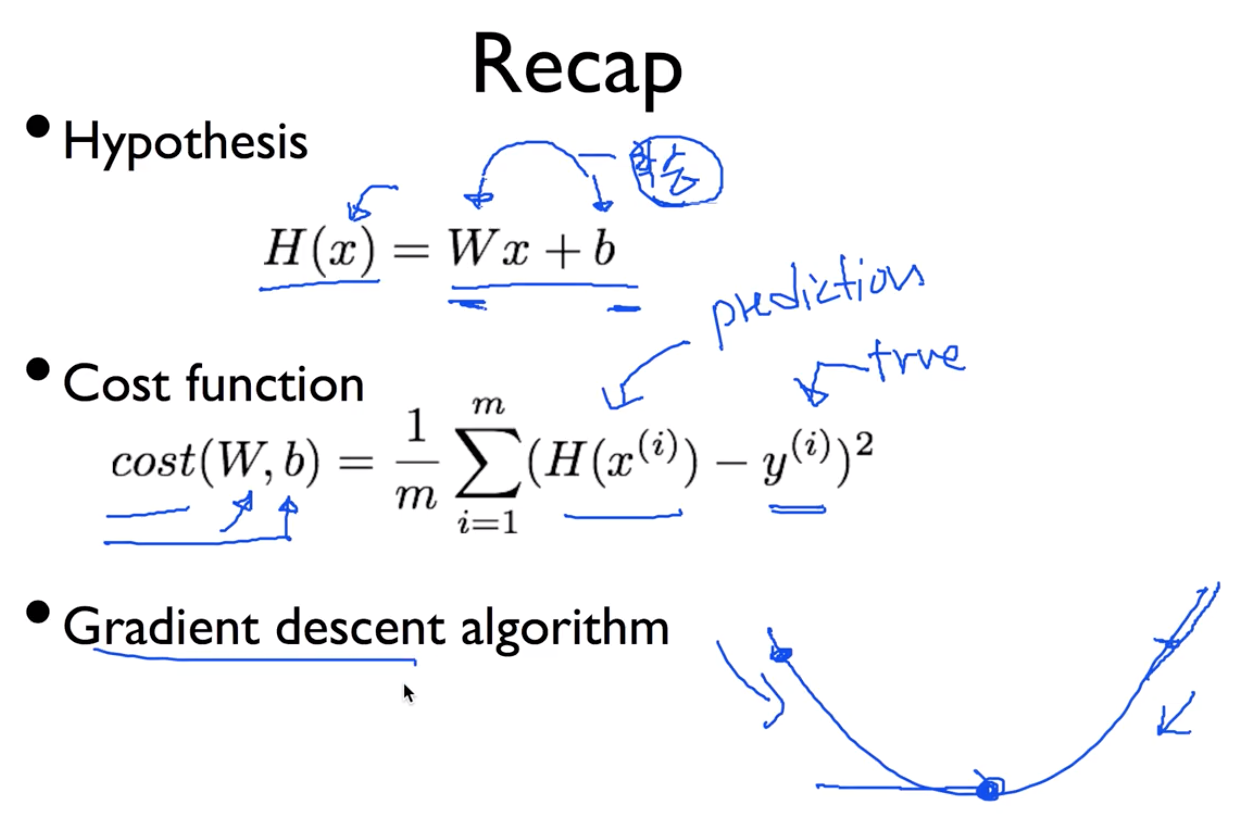
linear regression의 cost function이 convex function인지 확인해야 한다.

convex function은 언제나 답을 찾아준다.

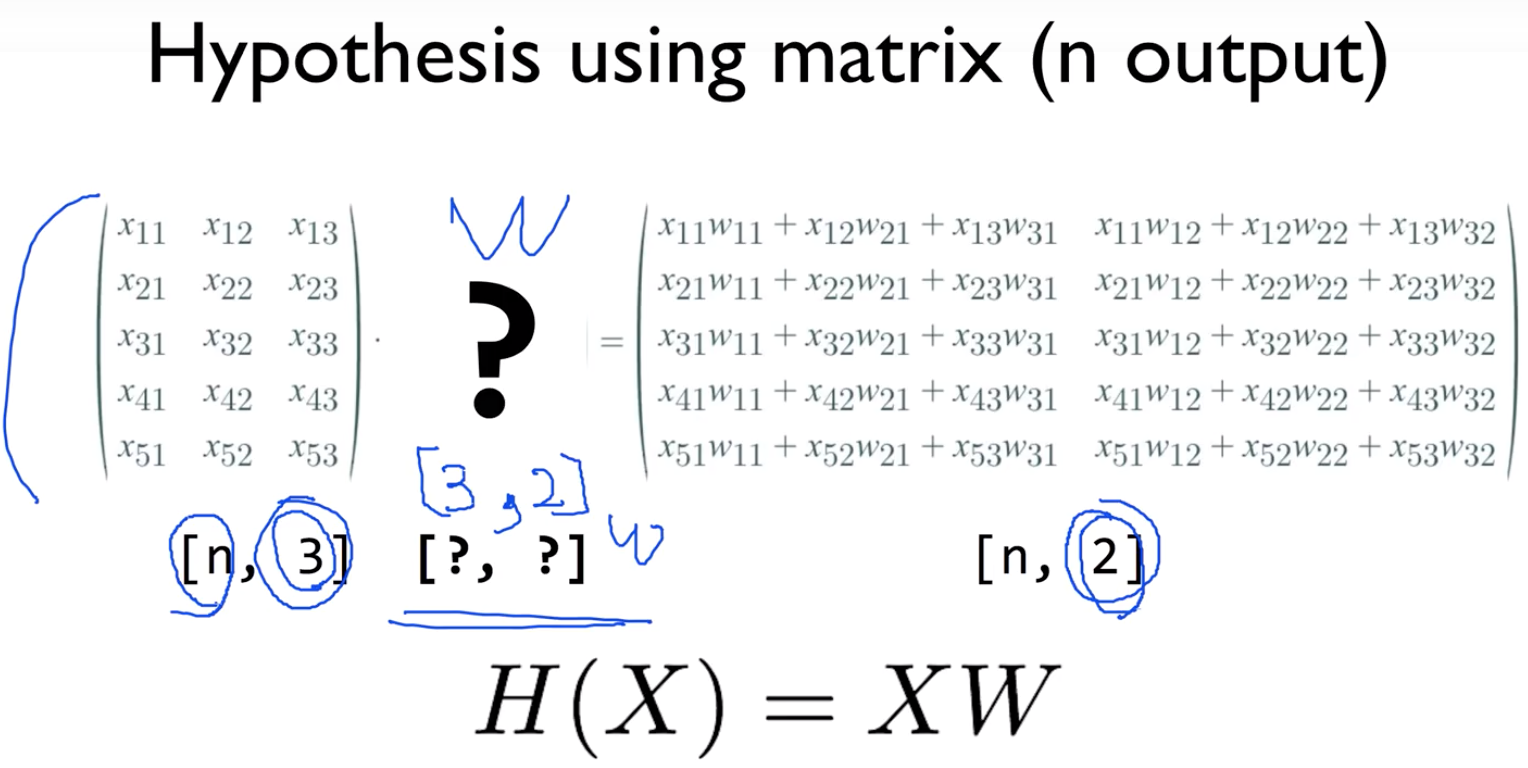
**섹션 4. 여러개의 입력(feature)의 Linear Regression**

**Multi-variable linear regression**

1. Recap



1. Hypothesis
   1. H(x1, x2, … , xn) = w1x1 + w2x2 + … + wnxn + b
   2. using matrix
      1. (x1 x2 x3) dot (w1 w2 w3)^T = (x1w1 + x2w2 + x3w3)
      2. **H(X) = XW**
         1. X = [# of instances(n), # of variables]
            1. n개의 instance를 텐서플로우에서는 None으로 표현
            2. numpy에서는 -1으로 표현
         2. W = [# of variables, y 갯수(?)]
         3. H = [# of instances(n), y 갯수(?)]

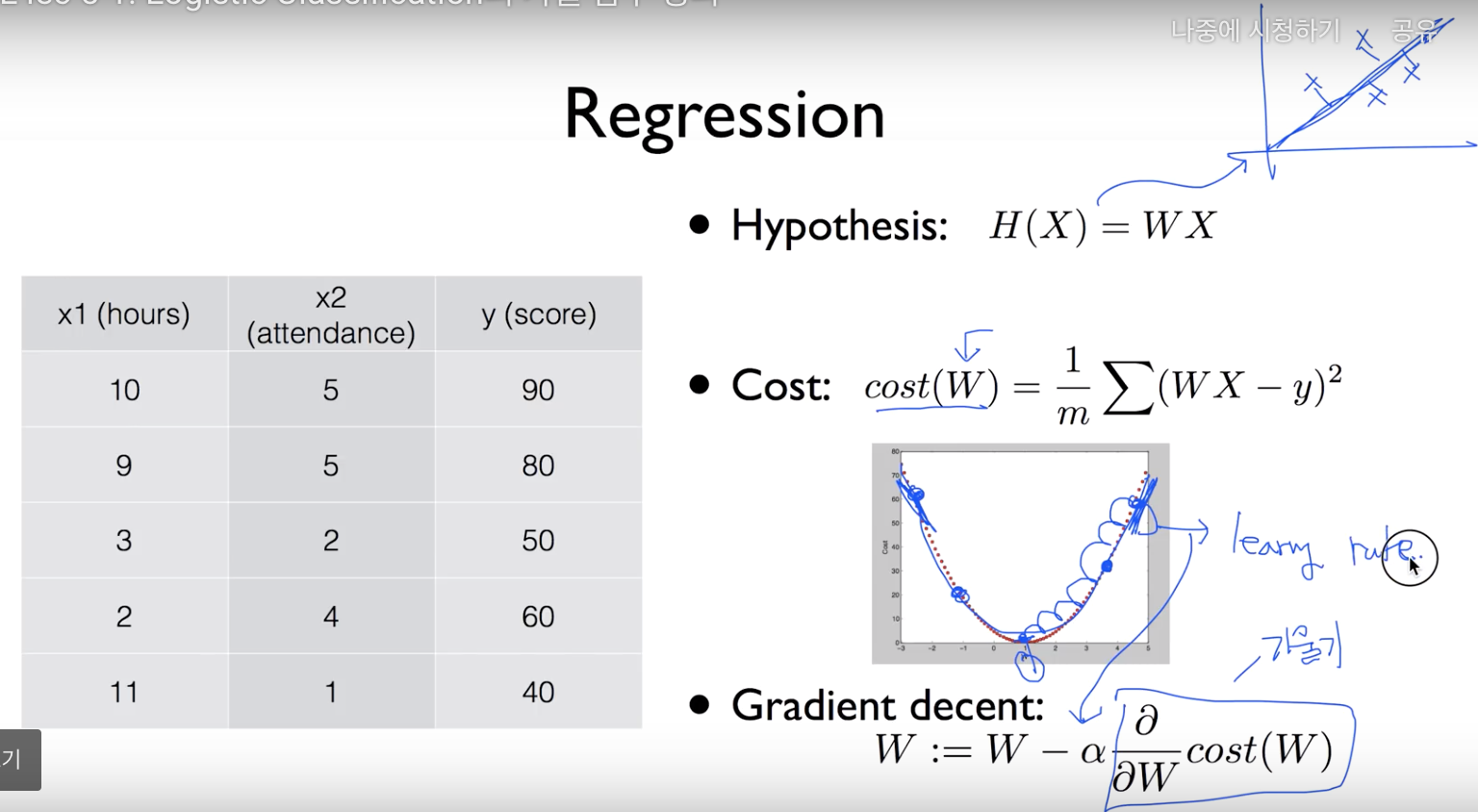


1. cost(W, b)
   1. = 1/m \* sum of [(H(x1(i), x2(i), …, xn(i)) - y(i)))^2] from i=1 to m
2. wx vs. xw
   1. Theory
      1. H(x) = Wx + b
   2. **Implementation (Tensorflow)**
      1. **H(X) = XW**

**섹션 5. Logistic (Regression) Classification**

**Logistic Classification의 가설 함수 정의**

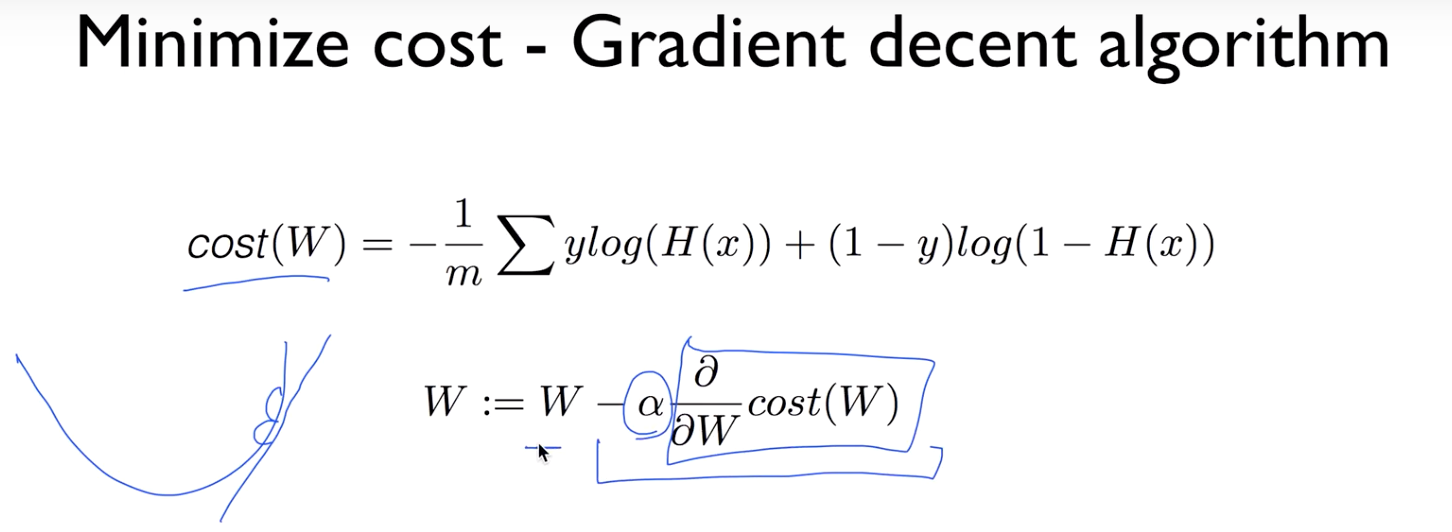
1. Regression
   1. 단순히 숫자를 예측



1. Binary Classification
   1. 둘 중 하나로 분류
   2. 0, 1 encoding
      1. spam detection: spam(1) or ham(0)
      2. facebook feed: show(1) or hide(0)
   3. 활용
      1. radiology: 나쁜 종양인지 좋은 종양인지
      2. finance: 투자할지 말지
   4. Pass(1)/Fail(0) based on study hours
2. z = H(x) = Wx + b
   1. Y is 0 or 1
   2. hypothesis can give values larger than 1 or less than 0
   3. **g(z) = 1/(1+e^(-z))**
      1. **sigmoid(logistic) function**
      2. H(x)를 0~1 사이의 값으로 만들어주는 함수
   4. => **z = WX, H(x) = g(z) = 1/(1+e^(-W(^T)X)**

**Logistic Regression의 cost 함수 설명**

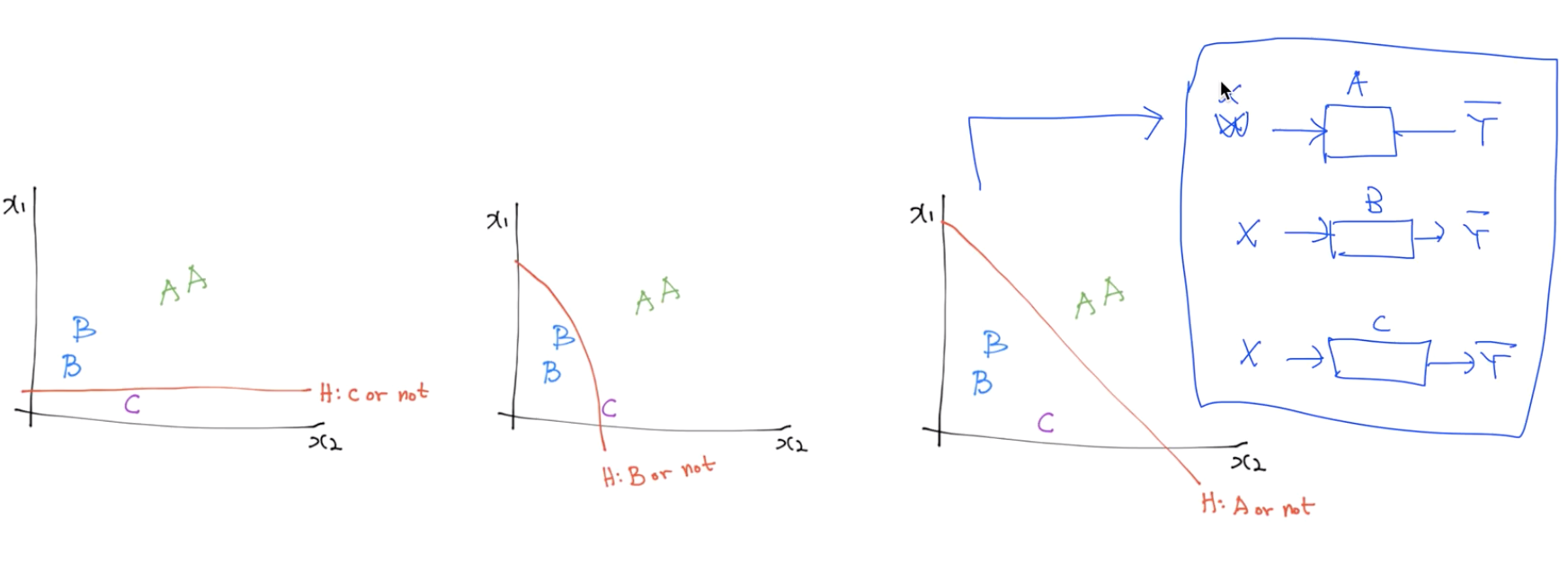
1. 기존 cost 함수를 사용할 때의 문제점
   1. local minimum이 아닌 global minimum을 찾아야 함
2. new cost function for logistic
   1. cost(W) = 1/m \* sum of C(H(x), y)
   2. **C(H(x), y) = -log(H(x)) if y=1, -log(1-H(x)) if y=0**
   3. **=> C(H(x), y) = ylog(H(x)) - (1-y)log(1-H(x))**
3. minimize cost - gradient descent algorithm

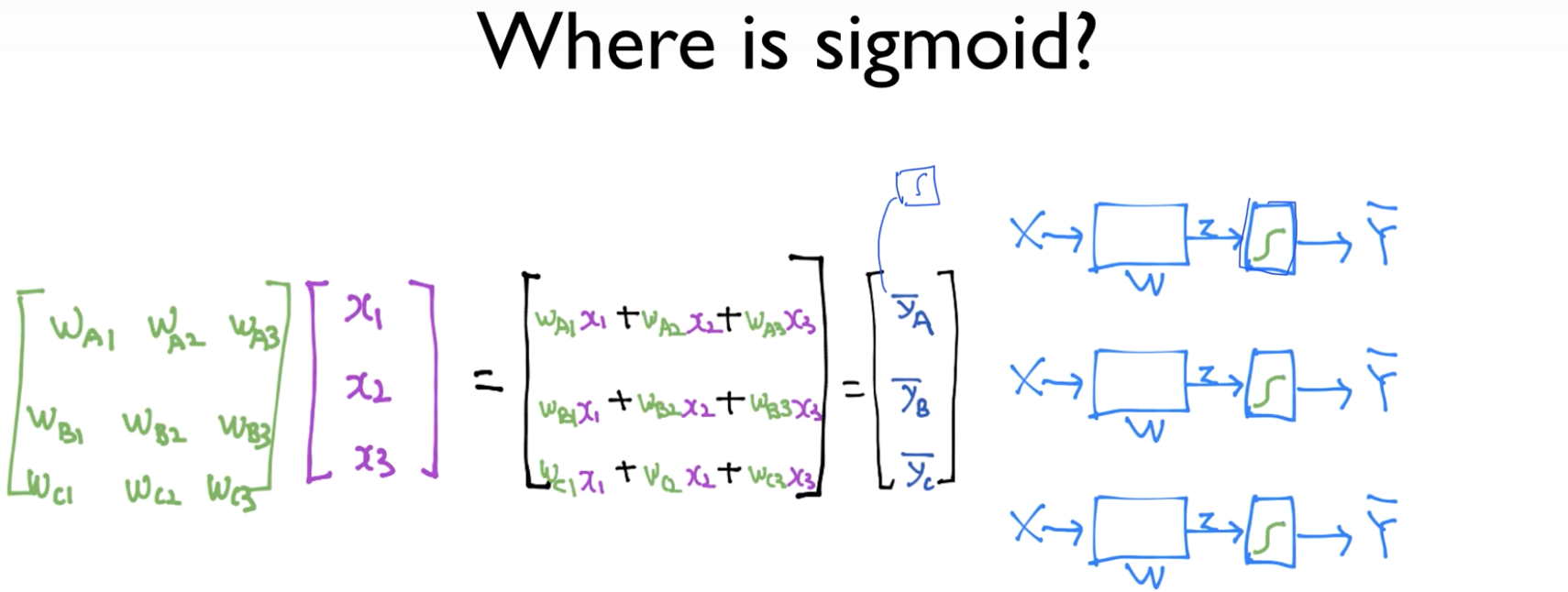


**섹션 6. Softmax Regression (Multinomial Logistic Regression)**

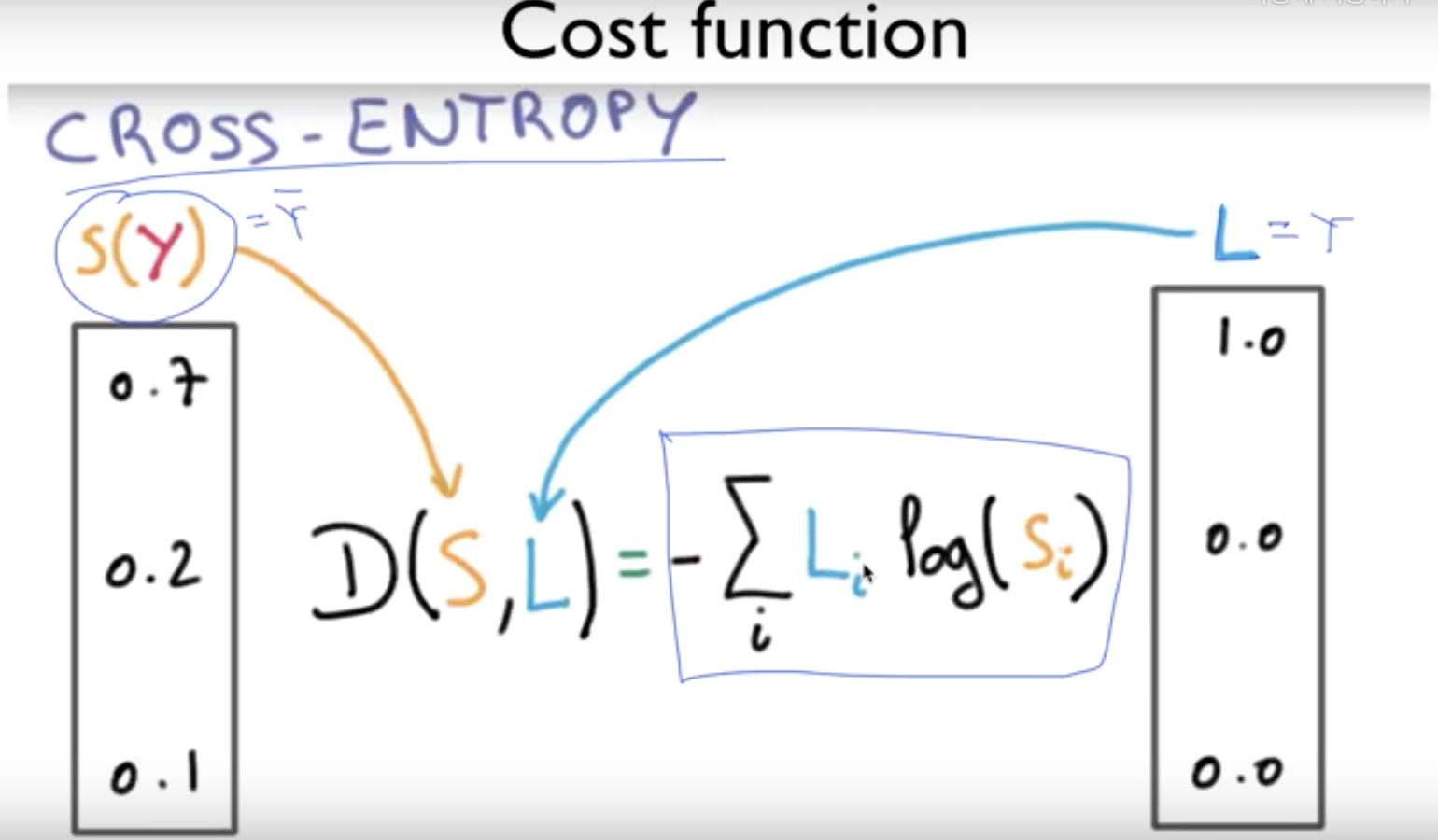
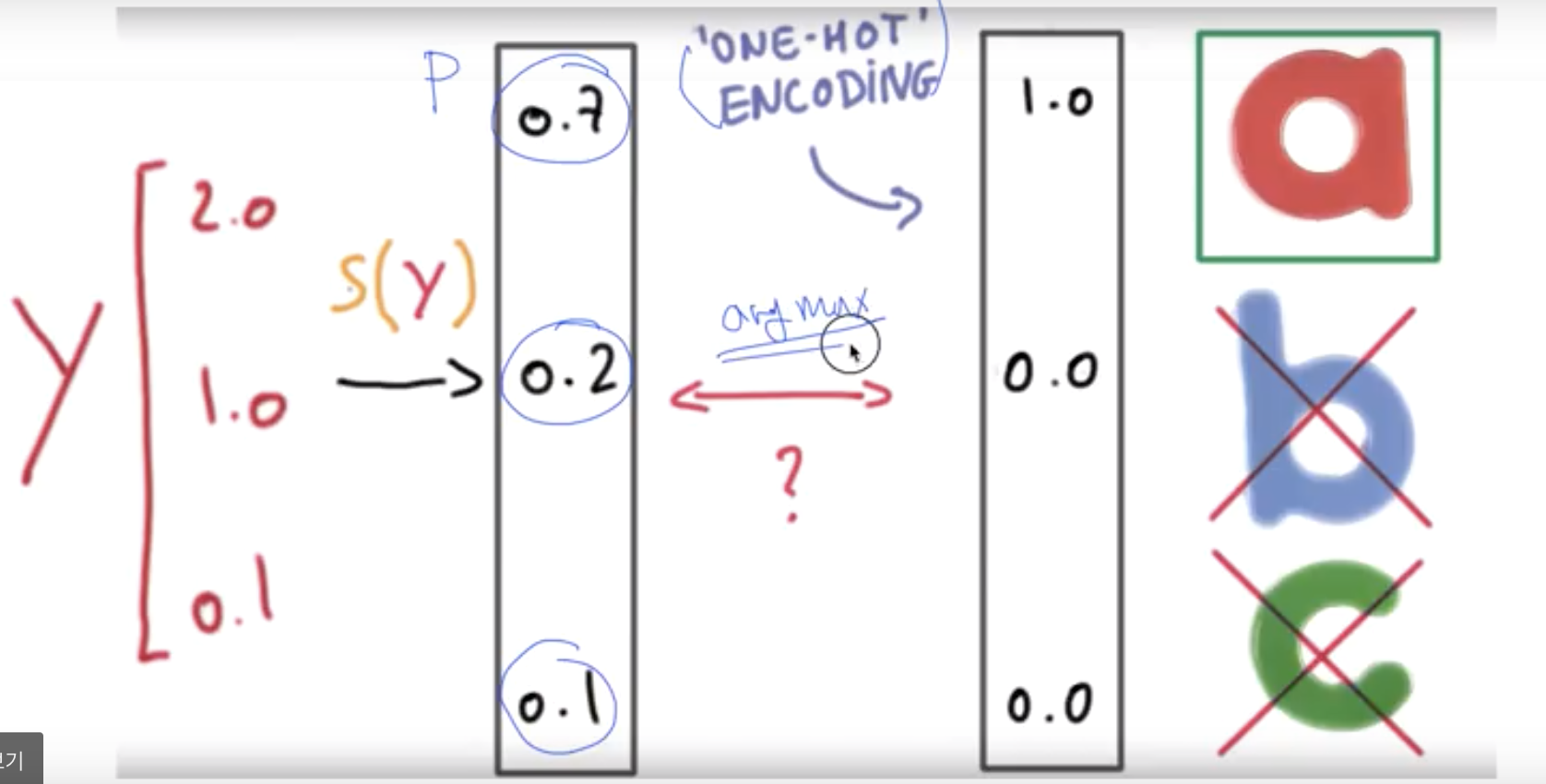
**Multinomial 개념 소개**

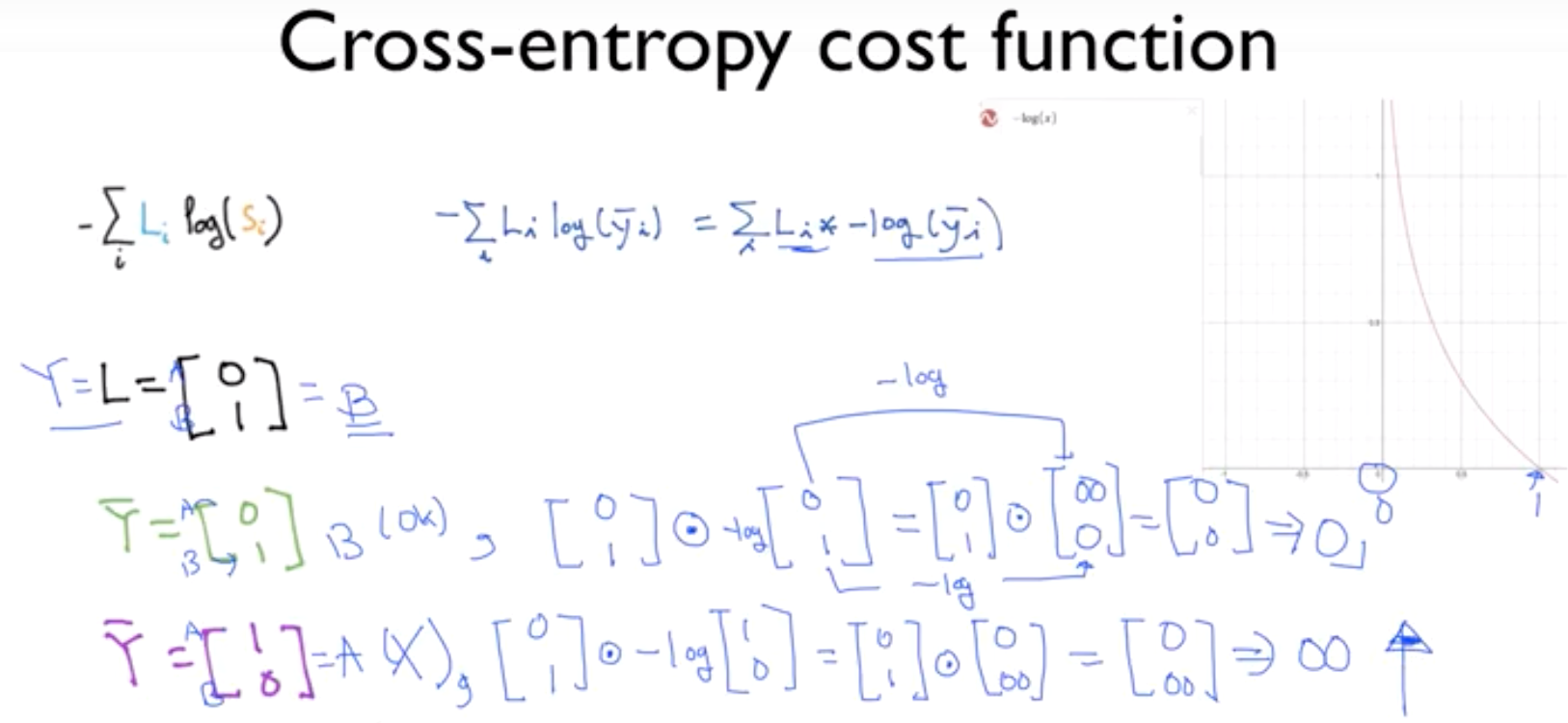
1. Binomial classification
   1. 0, 1로 구분
2. Multinomial classification
   1. n개의 예측이 존재할 때
   2. Binomial classification만으로도 Multinomial classification의 구현이 가능
      1. H: A or not
      2. H: B or not
      3. H: C or not
      4. =>A, B, C로 구분

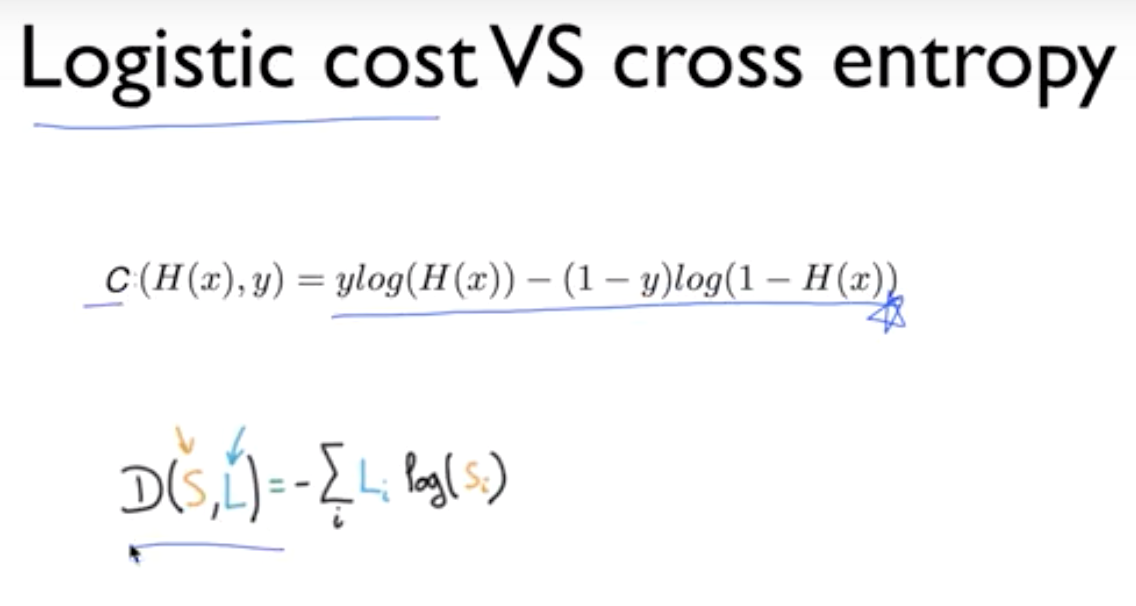


1. 

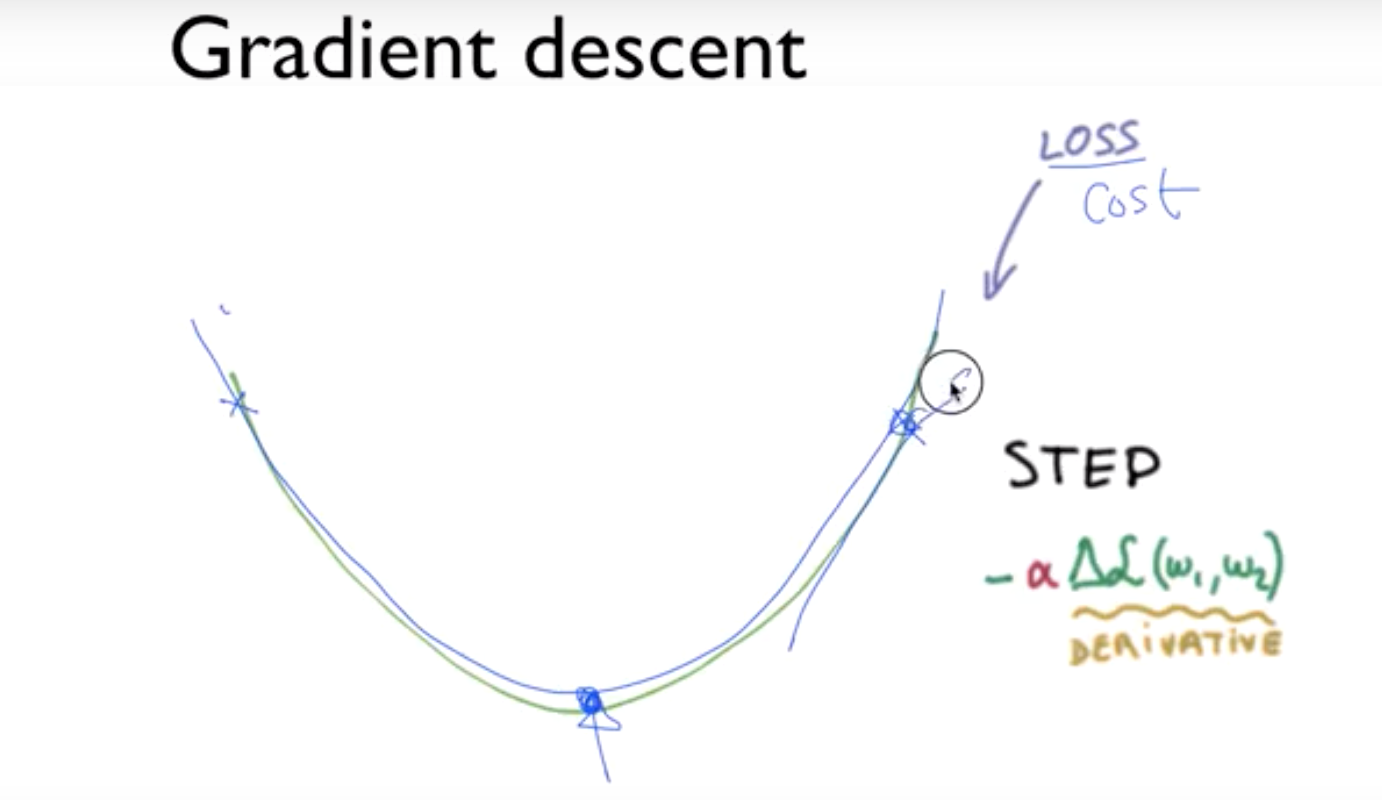
**Cost 함수 소개**

****

****

****

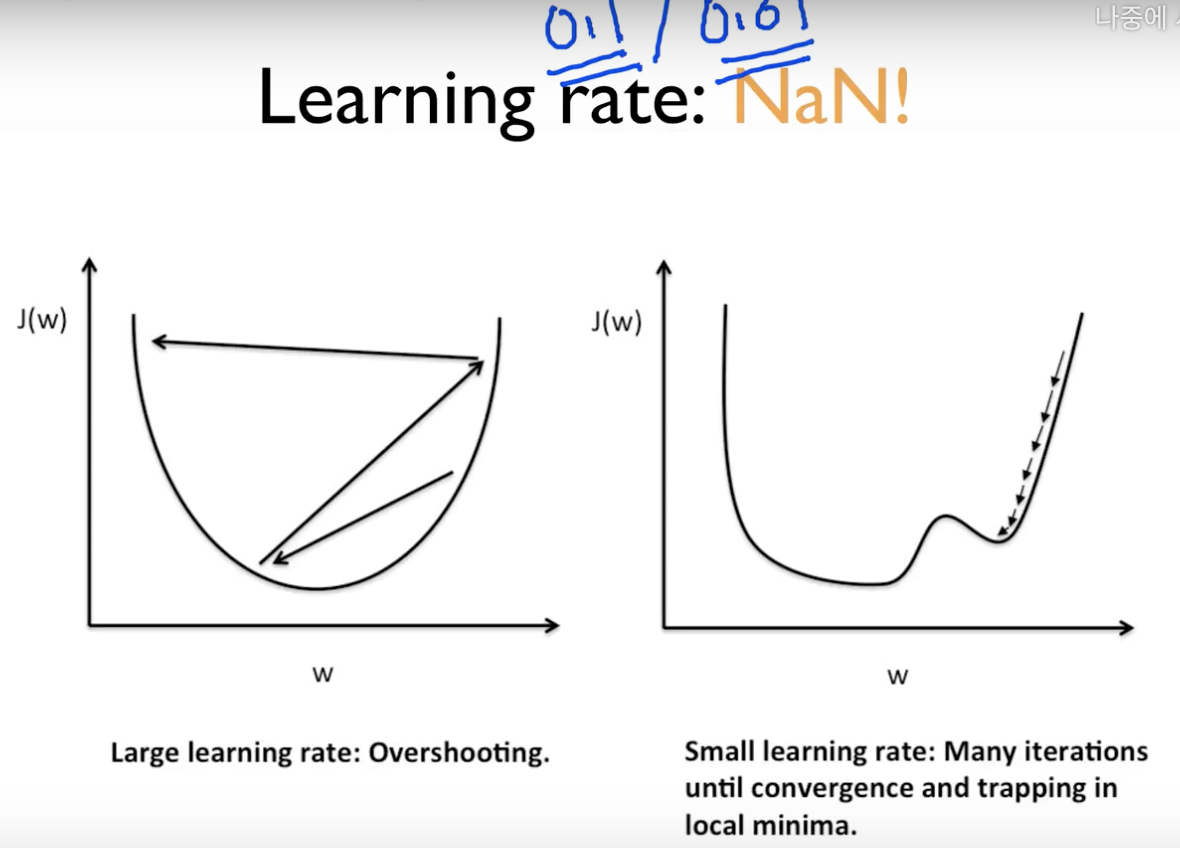
****

****

**섹션 7. ML의 실용과 몇 가지 팁**

**학습 rate, Overfitting, 그리고 일반화 (Regularization)**

1. Learning rate
   1. 굉장히 큰 값으로 정할 경우(매 step마다 굉장히 큰 간격으로 이동함)
      1. step이 너무 크면 바깥으로 튕겨나갈 수도 있음 (overshooting 발생)
         1. cost에 숫자가 아닌 값이 찍힘
   2. 너무 작을 경우
      1. 매 step마다 cost가 굉장히 작은 값으로 변화함dfew
      2. 해가 져도 바닥까지 하산하지 못 함
      3. 최저점이 아님에도 불구하고 stop되는 경향이 있음
         1. local minima에 빠져서 못 나옴
   3. => 0.01로 시작
      1. 발산될 경우 더 작은 값으로 조정
      2. 너무 느릴 경우 더 큰 값으로 조정



1. 데이터 값 차이가 너무 클 경우, 왜곡된 형태의 등고선이 나타남
   1. learning rate가 좋은 값이어도 튀어나가기 쉬움
   2. normalization 필요
      1. zero-centered data
         1. 데이터의 중심이 원점이 되도록 함
      2. normalized data
         1. 데이터가 항상 일정 범위 안에 들어가도록 함
      3. standardization
         1. (x - 평균) / 분산
         2. X\_std[:, 0] = (X[:, 0] - X[:, 0].mean()) / X[:, 0].std()
2. **Overfitting**
   1. 학습 데이터에 너무나 잘 맞는 모델을 만들게 될 경우
      1. 학습 데이터로 물어보면 답을 잘 하지만, 실제 데이터로 테스트를 진행하면 잘 맞지 않게 됨
   2. solutions
      1. training set을 늘리기
      2. feature의 갯수를 줄이기
      3. **regularization(일반화)**
         1. weight이 너무 크지 않도록 함
            1. weight이 너무 크면 구부러짐
            2. 구부리지 말고 좀 펴자 (weight을 좀 작은 값으로)

cost 식에 {lambda \* sum of (w^2)} term을 cost 식에 추가

regularization strength(lambda)

= 0: no regularization

big: regularization을 크게 고려함

cost를 최소화

**Training/Testing 데이타 셋**

“머신러닝 모델이 얼마나 잘 동작하는가(evaluation)”

1. Training set을 이용한 evaluation
   1. training set으로 학습시킨 후 동일한 training set으로 물어볼 경우
      1. 100% correct
      2. can memorize
      3. => 중간고사와 동일한 문제로 기말고사를 보는 셈 (X)
   2. training set과 test set으로 전체 original data set을 나누고, training set만을 이용해 학습시키기
      1. 이후 test set으로 평가
      2. training set(교과서), test set(실전)
         1. training set
            1. training set
            2. validation set

“모의 시험”을 통해 alpha, lambda 값을 튜닝하기 위함

* + - 1. testing set
  1. online learning
     1. training set이 100만 개이면, 한 번에 model에 넣지 않고 10만 개씩 잘라서 학습시키기
        1. 첫 번째 학습시킨 결과 + 추가 학습 => 두 번째 학습시킨 결과
        2. 두 번째 학습시킨 결과 + 추가 학습 => 세 번째 학습시킨 결과
        3. …
     2. 새로운 데이터가 추가되어도 이전에 있던 데이터를 새로 학습시키지 않고, 기존 데이터에 추가로 학습시킬 수 있음
  2. MNIST Dataset
     1. 미국에서 손으로 쓴우편번호를 분류하기 위해…

1. Accuracy
   1. How many of your predictions are correct?
   2. 최근 이미지 인식 정확도는 95%~99%