WITH TENSORFLOW

MACHINE LEARNING

MACHINE LEARNING

둘러보기

머신러닝의 확산 - 폭발적인 증가

- 무어의 법칙(Moore's Law)으로 컴퓨팅 비용이 급격히 낮아져 지금은 최소한의 비용으로 강력한 컴퓨팅 성능을 폭넓게 이용할 수 있다.
- 새롭고 혁신적인 알고리즘이 더욱 빠른 결과를 제공한다.
- 데이터 과학자들이 머신러닝을 효과적으로 적용하기 위한 이론과 실무 지식을 축적했다.

PROGRAMMING LANGUAGES - 나무위키

- R 언어
  현재 가장 많이 쓰이는 통계 기반 프로그래밍 언어이다. R 언어의 강점은 간단한 코딩으로 충분히 인지 가능할 정도로 시각화된 데이터를 얻어내는 것에 있다. 이를 통해 개발 파이프라인을 단축시키는데 많은 기여를 한다.

- 파이썬
  R 언어에 이어 이 분야에서 두 번째로 많이 쓰이는 언어이며, numpy 라이브러리를 써서 기계학습 알고리즘을 코딩하도록 도와준다. 비록 R 언어보다 코딩에 다소 시간이 걸리지만, 파이썬 언어의 대표적 장점인 이식성이 있어 다양한 분야에서 사용되어지며, Scipy 라이브러리 등을 추가하여 내부 변수의 계산을 하거나 Cython 등을 이용하여 알고리즘의 속도를 빠르게 하기에도 용이하다.

- Matlab
  수학적 정밀도가 어느 정도 보장되는 언어이다보니 주로 연구실 등에서 사용되어진다.

통계, 머신러닝, 데이터마이닝

- 통계 vs. 머신러닝
  - 통계는 분포나 가정을 사용해서 엄격한 규칙이 적용되는 설문조사나 실험 계획에 사용됨
  - 머신러닝은 대용량 데이터의 분석이나 패턴을 찾는데 사용됨
- 머신러닝 vs. 데이터마이닝
  - 머신러닝은 훈련 데이터를 통해 학습된 알려진 속성을 기반으로 한 예측에 중점
  - 데이터마이닝은 데이터의 미처 몰랐던 속성을 발견하는 것에 집중. 이는 데이터베이스의 지식 발견 부분의 분석 절차에 해당한다.
  - 이들은 방법적으로 중복되는 부분이 있다. 데이터마이닝에서 머신러닝은 필수가 아니지만, 머신러닝에서는 데이터마이닝이 필수라는 부분이 다르다.

인공지능, 머신러닝, 딥 러닝

- 인공지능
  - 외부 관찰자에게 인간처럼 스마트하게 소프트웨어를 작동시키는 폭넓은 방법, 알고리즘 및 기술
  - 머신러닝, 컴퓨터 비전, 자연어 처리, 로봇 공학 및 그와 관련된 모든 주제를 포괄하는 개념
- 머신러닝
  - 더 많은 데이터 축적을 통해 성능을 개선할 수 있도록 하는 다양한 알고리즘과 방법론
  - 신경망, 서포트 벡터 머신, 결정 트리, 베이지안 신뢰 네트워크, k 최근접 이웃, 자기 조직화 지도, 사례 기반 추론, 인스턴스 기반 학습, 은닉 마르코프 모델, 회귀 기법
- 딥 러닝
  - 신경망(**Neural Network**)을 부르는 다른 이름
  - 여러 개의 히든 레이어를 통해 깊게 학습한다고 해서 붙여진 이름

DEEP LEARNING

- 장점
  - 특징 간의 복잡한 상호작용을 탐지하는 능력
  - 최소한으로 처리된 원시 데이터에서 저수준의 특징을 학습하는 능력
  - 높은 기수(high-cardinality) 클래스 멤버십을 다루는 능력
  - 미분류(unlabeled) 데이터를 다루는 능력
- 결론
  - 다른 방법으로는 불가능한 유용한 결과 도출
  - 다른 방법보다 정확한 모델 구축
  - 다른 방법보다 구축 시간 단축

정의 – TOM M. MITCHELL, CARNEGIE MELLON UNIVERSITY

- A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.
- 태스크(T)에 대해 꾸준한 경험(E)을 통해 T에 대한 성능(P)을 높이는 것을 기계학습이라고 한다.
- 기계학습에서 가장 중요한 것은 E에 해당하는 데이터이다. 좋은 품질의 데이터를 많이 가지고 있다면 보다 높은 성능을 끌어낼 수 있다.

분류

- 지도 학습(Supervised Learning)
  - 회귀(Regresssion)
  - 분류(Classification)
- 비지도 학습(Unsupervised Learning)
  - 군집화(Clustering)
  - 분포 추정(Underlying Probability Density Estimation)
- 강화 학습(Reinforcement Learning)

둘러보기

- 예측
  Linear regression, Regression tree, Kernel regression, Support vector regression
- 분류
  Logistic regression, Decision tree, Nearest-neighbor classifier, Kernel discriminate analysis, Neural network, Support Vector Machine, Random forest, Boosted tree
- 차원(변수) 축소
  Principal component analysis, Non-negative matrix factorization, Independent component analysis, Manifold learning, SVD
- 그룹화
  k-means, Hierarchical clustering, mean-shift, self-organizing maps(SOMs)
- 선행학습(Pre-training), 2차분류
  Deep Learning(Stacked Restricted Boltzmann Machine, Stacked Auto-Encoders등을 사용한 Multi layers Neural Nets, Non-linear Transformation)
- 데이터 비교
  Bipartite cross-matching, n-point correlation two-sample testing, minimum spanning tree

SUPERVISED LEARNING (1)

- 서포트 벡터 머신 (support vector machine)
  패턴 인식, 자료 분석을 위한 지도 학습 모델이며, 주로 분류와 회귀 분석을 위해 사용한다. 두 카테고리 중 어느 하나에 속한 데이터의 집합이 주어졌을 때, SVM 알고리즘은 주어진 데이터 집합을 바탕으로 하여 새로운 데이터가 어느 카테고리에 속할지 판단하는 비확률적 이진 선형 분류 모델을 만든다. 만들어진 분류 모델은 데이터가 사상된 공간에서 경계로 표현되는데 SVM 알고리즘은 그 중 가장 큰 폭을 가진 경계를 찾는 알고리즘이다.

- 은닉 마르코프 모델 (Hidden Markov model)
  통계적 마르코프 모델의 하나로, 시스템이 은닉된 상태와 관찰가능한 결과의 두 가지 요소로 이루어졌다고 보는 모델이다. 관찰 가능한 결과를 야기하는 직접적인 원인은 관측될 수 없는 은닉 상태들이고, 오직 그 상태들이 마르코프 과정을 통해 도출된 결과들만이 관찰될 수 있기 때문에 '은닉'이라는 단어가 붙게 되었다. 음성 인식, 필기 인식, 동작 인식, 품사 태깅, 악보에서 연주되는 부분을 찾는 작업, 부분 방전, 생물정보학과 같이 시간의 영향을 받는 시스템의 패턴을 인식하는 작업에 유용한 것으로 알려져있다.

SUPERVISED LEARNING (2)

- 회귀 분석 (Regression)
  통계학에서, 회귀분석(regression analysis)은 관찰된 연속형 변수들에 대해 두 변수 사이의 모형을 구한뒤 적합도를 측정해 내는 분석 방법이다. 회귀분석은 시간에 따라 변화하는 데이터나 어떤 영향, 가설적 실험, 인과 관계의 모델링등의 통계적 예측에 이용될 수 있다.

- 신경망 (Neural network)
  인공신경망은 생물학의 신경망(특히 뇌)에서 영감을 얻은 통계학적 학습 알고리즘이다. 인공신경망은 시냅스의 결합으로 네트워크를 형성한 인공 뉴런(노드)이 학습을 통해 시냅스의 결합 세기를 변화시켜, 문제 해결 능력을 가지는 모델 전반을 가리킨다. 좁은 의미에서는 역전파법을 이용한 다층 퍼셉트론을 가리키는 경우도 있지만, 인공신경망은 이에 국한되지 않는다.

- 나이브 베이즈 분류 (Naive Bayes Classification)
  특성들 사이의 독립을 가정하는 베이즈 정리를 적용한 확률 분류기의 일종으로 1950 년대 이후 광범위하게 연구되고 있다. 텍스트 분류에 사용됨으로써 문서를 여러 범주 (예: 스팸, 스포츠, 정치)중 하나로 판단하는 문제에 대한 대중적인 방법으로 남아있다.

UNSUPERVISED LEARNING

- 클러스터링(Clustering)
  개체를 다수의 메트릭스에서 상호 유사한 세그먼트 또는 클러스터로 그룹화하는 기법. 고객 세분화가 클러스터링의 실제 예다. 클러스터링 알고리즘은 무척 다양한데, 가장 널리 사용되는 것이 k-평균(k-means)이고 비슷한 알고리즘으로 Gaussian Mixture Model도 있다.
- 독립 성분 분석(Independent Component Analysis)
  다변량의 신호를 통계적으로 독립적인 하부 성분으로 분리하는 계산 방법이다. 각 성분은 비가우스 성 신호로서 서로 통계적 독립을 이루는 성분으로 구성되어 있다. 독립 성분 분석은 블라인드 신호를 분리하는 특별한 방법이다.
  독립 성분 분석의 전형적인 알고리즘은 복잡성을 줄이기 위한 전 단계로서 중심화 (centering), 백색화 (whitening), 차원 감소 (dimensionality reduction) 등의 과정이 필요하다. 백색화와 차원 감소는 주 성분 분석 (Principal Component Analysis)과 특이값 분해 (Singular Value Decomposition)로 한다.

REINFORCEMENT LEARNING

- 개요
  - 환경을 탐색하는 에이전트가 현재의 상태를 인식하여 어떤 행동을 취함
  - 에이전트는 환경으로 부터 포상을 얻음
    - 포상은 양수와 음수 둘 다 가능
  - 에이전트가 앞으로 누적될 포상을 최대화 하는 정책을 찾는 방법
- 강화 학습과 지도 학습의 차이점
  - 지도 학습 : 알고 있는 지식을 이용해 원하지 않는 행동을 명시적으로 수정하며 모델 업데이트
  - 강화 학습 : 알고 있는 지식과 아직 조사되지 않는 영역을 탐험하는 것 사이의 균형을 잡는 것
- Multi-armed Bandit 문제, Q-Learning 알고리즘

머신러닝의 실제 사용 사례 (1)

- 사기 방지: 1억 5,000만 개의 디지털 월릿을 통해 연간 2,000억 달러 이상의 결제를 처리하는 페이팔(PayPal)은 온라인 결제업계의 선두 주자다. 이 정도 규모에서는 사기 비율이 낮다 해도 그 비용은 상당하다. 창업 초기에는 월별 사기 피해 금액이 1,000만 달러에 이르렀다.
- 타겟팅 디지털 디스플레이: 광고 기술 기업 Dstillery는 실시간 입찰 플랫폼에서 타겟 디지털 디스플레이 광고를 진행하도록 한다. 개인의 브라우징 내역, 방문, 클릭 및 구매에 대해 수집된 데이터를 사용해 한 번에 수백 개의 광고 캠페인을 처리하며 초당 수천 건의 예측을 실행한다.
- 콘텐츠 추천: Comcast는 TV 서비스 고객을 위해 각 고객의 이전 시청 습관을 기반으로 한 실시간으로 개인 맞춤화된 콘텐츠를 추천한다. 수십억 개의 내역 기록을 사용해 각 고객별로 고유한 취향 프로필을 작성한 다음, 공통적인 취향을 가진 고객을 클러스터로 묶는다. 그런 후 각 고객 클러스터를 대상으로 가장 인기있는 콘텐츠를 실시간으로 추적 및 추천한다.

머신러닝의 실제 사용 사례 (2)

- 자동차 품질 개선: Jaguar Land Rover의 신형 차량에는 60개의 온보드 컴퓨터가 탑재되며 이 컴퓨터는 2만 개 이상의 메트릭스를 기준으로 매일 1.5GB의 데이터를 생성한다. 고객이 차량을 실제로 어떻게 다루는 지를 파악해서 얻은 정확한 사용 데이터를 통해 설계자는 부품 고장과 잠재적 안전위험을 예측할 수 있다. 조건에 맞는 차량 엔지니어링에도 도움이 된다.
- 유망 잠재 고객에 집중: 마케터들은 최적의 판매와 마케팅 기회, 그리고 최적의 제품을 판단하기 위한 도구로 구매 성향 모델을 사용한다. 라우터부터 케이블 TV 박스에 이르기까지 방대한 제품을 보유한 Cisco의 마케팅 분석팀은 몇 시간 만에 6만 개의 모델을 교육시키고 1억 6,000만 명의 잠재 고객을 확보했다.
- 의료 보건 서비스 개선: 환자의 재입실은 의료 보험 공단과 민간 보험사가 재입실 비율이 높은 병원에 불이익을 줄 수 있다. 건강한 상태를 유지할 가능성이 충분히 높은 환자만 퇴원시키는 역량이 병원의 재무에 큰 영향을 미치게 된다. Carolinas Healthcare System은 환자의 위험 점수를 계산하고 병원 사례 관리자는 이를 바탕으로 퇴원 결정을 내린다.

MACHING LEARNING

용어 정리

1. 회귀분석 (HTTP://MATH7.TISTORY.COM/118 에서 발췌)

- 점들이 퍼져있는 형태에서 패턴을 찾아내고,
  이 패턴을 활용해서 무언가를 예측하는 분석.

- 새로운 표본을 뽑았을 때 평균으로 돌아가려는 특징이 있기 때문에 붙은 이름

- 회귀(回歸 돌 회, 돌아갈 귀)라는 용어는 일반적으로 '돌아간다'는 정도로만 사용하기 때문에 회귀로부터 '예측'이라는 단어를 떠올리기는 쉽지 않다.

2. LINEAR REGRESSION

- 2차원 좌표에 분포된 데이터를 1차원 직선 방정식을 통해 표현되지 않은 데이터를 예측하기 위한 분석 모델.
- 머신러닝 입문에서는 기본적으로 2차원이나 3차원까지만 정리한다.
- 여기서는 편의상 1차원 직선으로 정리하고 있다. xy축 좌표계에서 직선을 그렸다고 생각하면 된다.

3. HYPOTHESIS

- **Linear Regression**에서 사용하는 **1**차원 방정식을 가리키는 용어로, 우리말로는 가설이라고 한다. 수식에서는 **h(x)** 또는 **H(x)**로 표현된다.

- 최저점(**minimize cost**)이라는 정답을 찾기 위한 가정이기 때문에 가설이라고 부를 수 있다.

- **H(x) = Wx + b**
  ==> x에 대한 **1**차 방정식

4. COST (비용)

- 앞에서 설명한 **Hypothesis** 방정식에 대한 비용(**cost**)으로 방정식의 결과가 크게 나오면 좋지 않다고 얘기하고 루프를 돌 때마다 **W**와 **b**를 비용이 적게 발생하는 방향으로 수정하게 된다.
- 미분을 사용해서 스스로 최저 비용을 찾아간다.
- **Gradient Descent Algorithm**을 사용해서 최저 비용을 찾는다.

5. COST 함수

- **Hypothesis** 방정식을 포함하는 계산식
- 현재의 기울기(**W**)와 절편(**b**)에 대해 비용을 계산해 주는 함수
- **W**와 **b**가 변함에 따라 반드시 **convex**(오목)한 형태로 설계되어야 하는 것이 핵심.
- **convex**하지 않다면, 경사를 타고 내려갈 수 없기 때문에 최저점 계산이 불가능해질 수 있다.
- **Linear Regression**을 비롯한 머신러닝 전체에서
  최소 비용을 검색하기 위한 역할 담당

6. GRADIENT DESCENT ALGORITHM

- 딥러닝의 핵심 알고리즘
- 경사타고 내려가기, 경사하강법 등의 여러 용어로 번역되었다.
- 미분을 사용해서 비용이 작아지는 방향으로 진행하는 알고리즘
- 생각보다 어렵지 않고 간단한 미분 정도만 이해하면 알고리즘 자체는 너무 단순하다.
- 텐서플로우에 포함된 **Optimizer**는 대부분 **Gradient Descent Algorithm**에서 파생된 방법을 사용하고 있다.

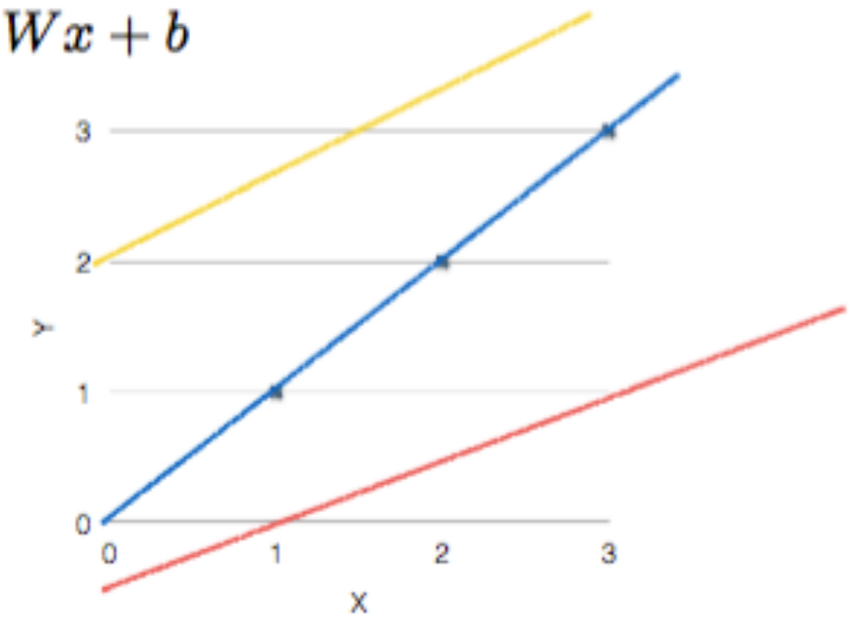
MACHINE LEARNING

LINEAR REGRESSION

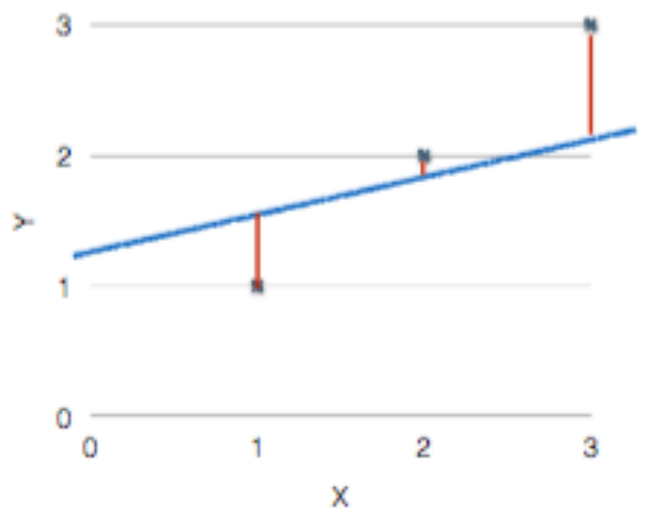
좋은 가설?

(Linear) Hypothesis

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



Which hypothesis is better?



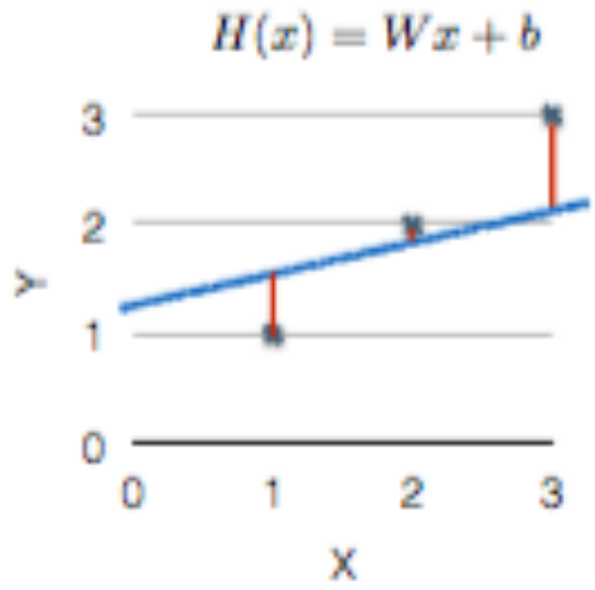
직선으로부터 점까지의 거리 계산

Cost function

- How fit the line to our (training) data

$$\frac{(H(x^{(1)}) - y^{(1)})^2 + (H(x^{(2)}) - y^{(2)})^2 + (H(x^{(3)}) - y^{(3)})^2}{3}$$

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



COST 수식

Cost function

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

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

$$cost(W, b) = \frac{1}{m}\sum_{i=1}^{m}(H(x^{(i)}) - y^{(i)})^2$$

최저 COST를 만드는 W와 B는?

- H(x) = wx + b
- x = [1, 2, 3]
  y = [1, 2, 3]
- w와 b가 아래와 같다면?
  1. w = 1/2, b = 0
  2. w = 0, b = 2
  3. w = 1, b = 1/2
  4. w = 1, b = 1

THE GOAL OF LINEAR REGRESSION

- Cost를 최소로 만드는
  W(Weight)와 b(bias)를 찾는 것.
- Linear Regression의 목표는 곧 머신러닝의 목표!

MACHINE LEARNING

GRADIENT DESCENT ALGORITHM

HOW TO HIDE BIAS?

Hypothesis and Cost

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

$$cost(W, b) = \frac{1}{m}\sum_{i=1}^{m}(H(x^{(i)}) - y^{(i)})^2$$

Simplified hypothesis

$$H(x) = Wx$$

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

COST FUNCTION GRAPH

What *cost(W)* looks like?

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

X	Y
1	1
2	2
3	3

- W=1, cost(W)=0

  $\frac{1}{3}((1*1-1)^2 + (1*2-2)^2 + (1*3-3)^2)$

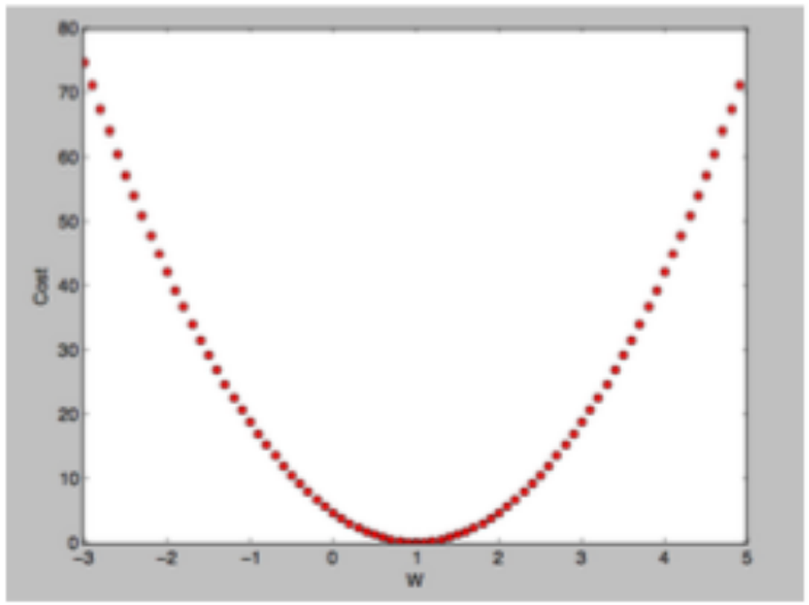
- W=0, cost(W)=4.67

  $\frac{1}{3}((0*1-1)^2 + (0*2-2)^2 + (0*3-3)^2)$

- W=2, cost(W)=?

How to minimize cost?

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



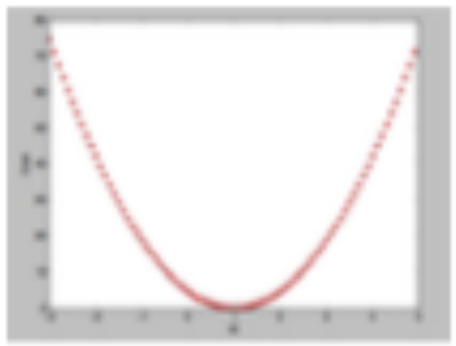
GRADIENT DESCENT ALGORITHM

Gradient descent algorithm

- Minimize cost function
- Gradient descent is used many minimization problems
- For a given cost function, *cost (W, b)*, it will find *W, b* to minimize cost
- It can be applied to more general function: *cost (w1, w2, …)*

HOW IT WORKS?

How it works?



- Start with initial guesses
  - Start at 0,0 (or any other value)
  - Keeping changing W and b a little bit to try and reduce cost(W, b)
- Each time you change the parameters, you select the gradient which reduces cost(W, b) the most possible
- Repeat
- Do so until you converge to a local minimum
- Has an interesting property
  - Where you start can determine which minimum you end up

FORMAL DEFINITION

Formal definition

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

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

Formal definition

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

$$W := W - \alpha\frac{\partial}{\partial W}cost(W)$$

Formal definition

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

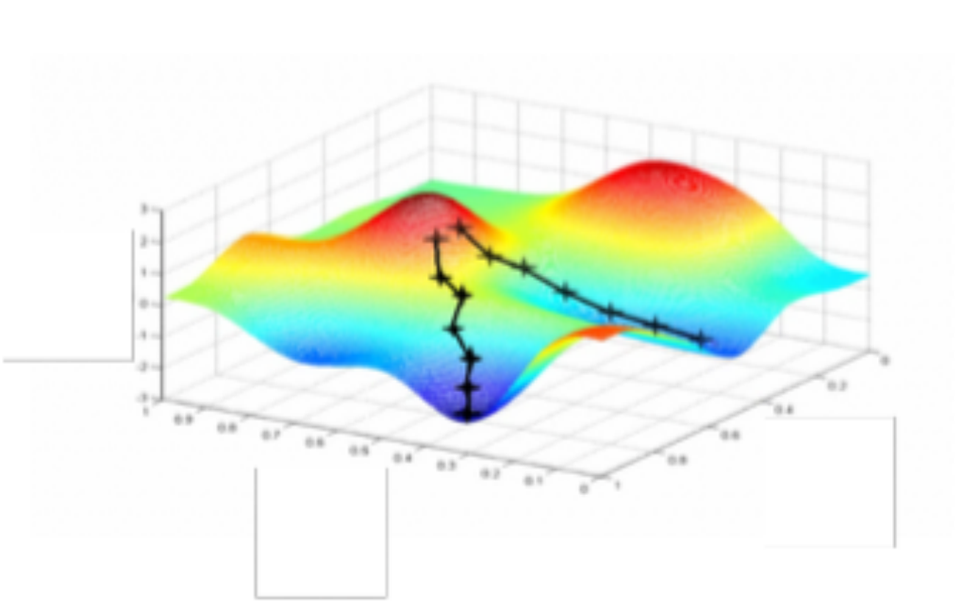
$$W := W - \alpha\frac{1}{2m}\sum_{i=1}^{m}2(Wx^{(i)} - y^{(i)})x^{(i)}$$

$$W := W - \alpha\frac{1}{m}\sum_{i=1}^{m}(Wx^{(i)} - y^{(i)})x^{(i)}$$

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

CONVEX FUNCTION

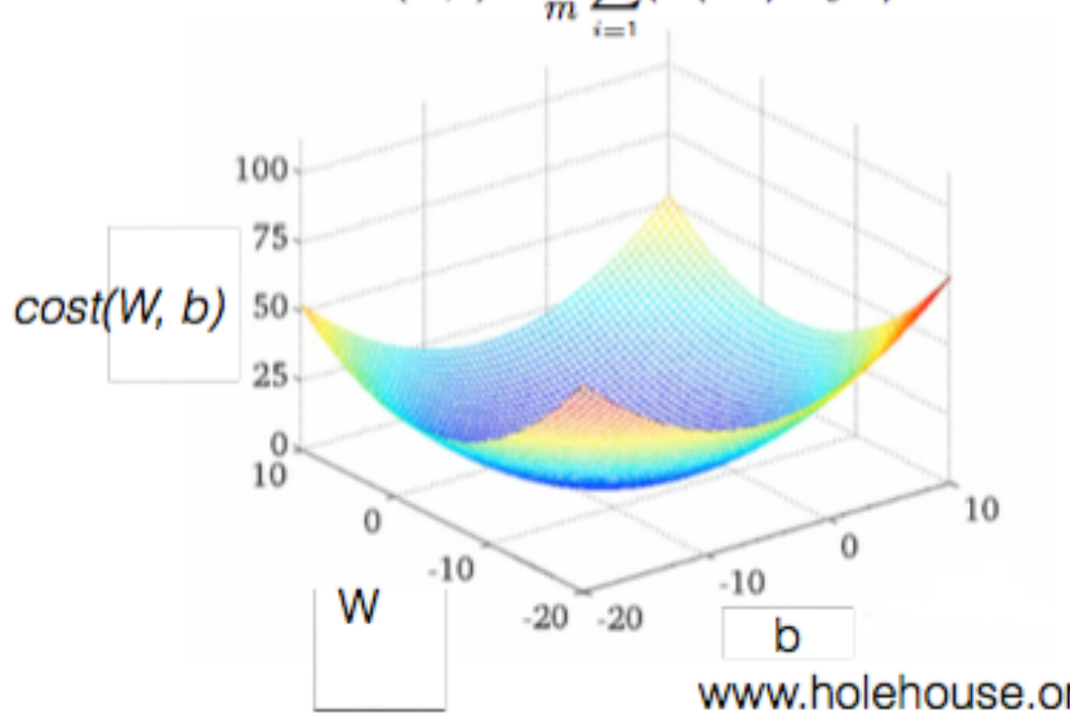
Convex function



www.holehouse.org/mlclass/

Convex function

$$cost(W,b) = \frac{1}{m}\sum_{i=1}^{m}(H(x^{(i)}) - y^{(i)})^2$$



www.holehouse.org/mlclass/

MACHINE LEARNING

MULTI-VARIABLE LINEAR REGRESSION

MULTI FEATURES

one-variable
one-feature

x (hours)	y (score)
10	90
9	80
3	50
2	60
11	40

multi-variable/feature

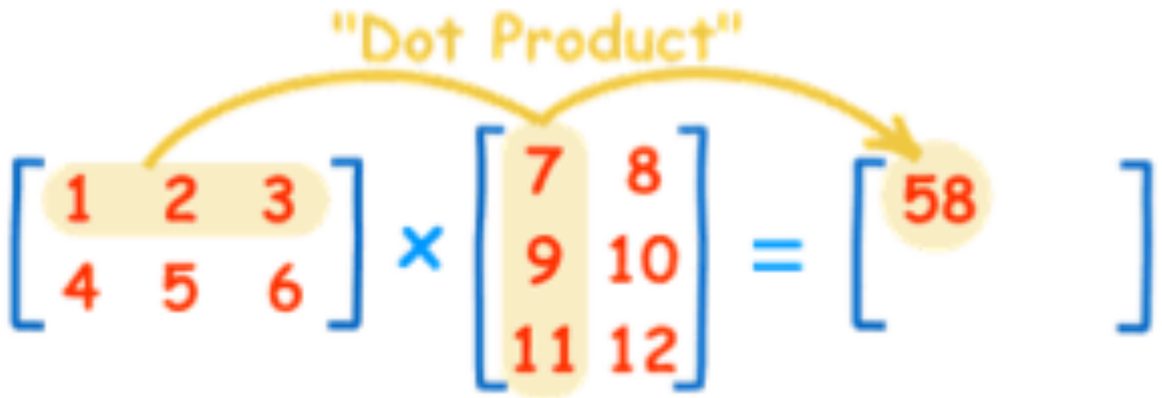
x1 (hours)	x2 (attendance)	y (score)
10	5	90
9	5	80
3	2	50
2	4	60
11	1	40

HYPOTHESIS AND COST FUNCTION

Hypothesis	Cost function
$H(x) = Wx + b$	$H(x_1, x_2) = w_1x_1 + w_2x_2 + b$
$H(x_1, x_2) = w_1x_1 + w_2x_2 + b$	$cost(W, b) = \frac{1}{m}\sum_{i=1}^{m}(H(x^{(i)}) - y^{(i)})^2$

MATRIX AND TRANSPOSE

Matrix multiplication



Transpose

$$\begin{bmatrix} 6 & 4 & 24 \\ 1 & -9 & 8 \end{bmatrix}^T = \begin{bmatrix} 6 & 1 \\ 4 & -9 \\ 24 & 8 \end{bmatrix}$$

MATRIX AND HYPOTHESIS

Matrix

$$w_1 x_1 + w_2 x_2 + w_3 x_3 + \ldots + w_n x_n$$

$$\begin{bmatrix} w1 & w2 & w3 \end{bmatrix} \times \begin{bmatrix} x1 \\ x2 \\ x3 \end{bmatrix} = \begin{bmatrix} w1 \times x1 + w2 \times x2 + w3 \times x3 \end{bmatrix}$$

Hypothesis

$$\begin{bmatrix} w1 & w2 & w3 \end{bmatrix} \times \begin{bmatrix} x1 \\ x2 \\ x3 \end{bmatrix} = \begin{bmatrix} w1 \times x1 + w2 \times x2 + w3 \times x3 \end{bmatrix}$$

$$H(X) = WX + b$$

HYPOTHESIS WITHOUT B

Hypothesis without *b*

$$\begin{bmatrix} b & w1 & w2 & w3 \end{bmatrix} \times \begin{bmatrix} 1 \\ x1 \\ x2 \\ x3 \end{bmatrix} = \begin{bmatrix} b \times 1 + w1 \times x1 + w2 \times x2 + w3 \times x3 \end{bmatrix}$$

$$H(X) = WX$$

HYPOTHESIS USING TRANSPOSE

W vs X

$$\begin{bmatrix} b & w1 & w2 & w3 \end{bmatrix} \times \begin{bmatrix} 1 \\ x1 \\ x2 \\ x3 \end{bmatrix} = \begin{bmatrix} b \times 1 + w1 \times x1 + w2 \times x2 + w3 \times x3 \end{bmatrix}$$

$$H(X) = WX$$

Hypothesis using Transpose

$$\begin{bmatrix} b & w1 & w2 & w3 \end{bmatrix} \times \begin{bmatrix} 1 \\ x1 \\ x2 \\ x3 \end{bmatrix} = \begin{bmatrix} b \times 1 + w1 \times x1 + w2 \times x2 + w3 \times x3 \end{bmatrix}$$

$$H(X) = W^T X$$

MACHINE LEARNING

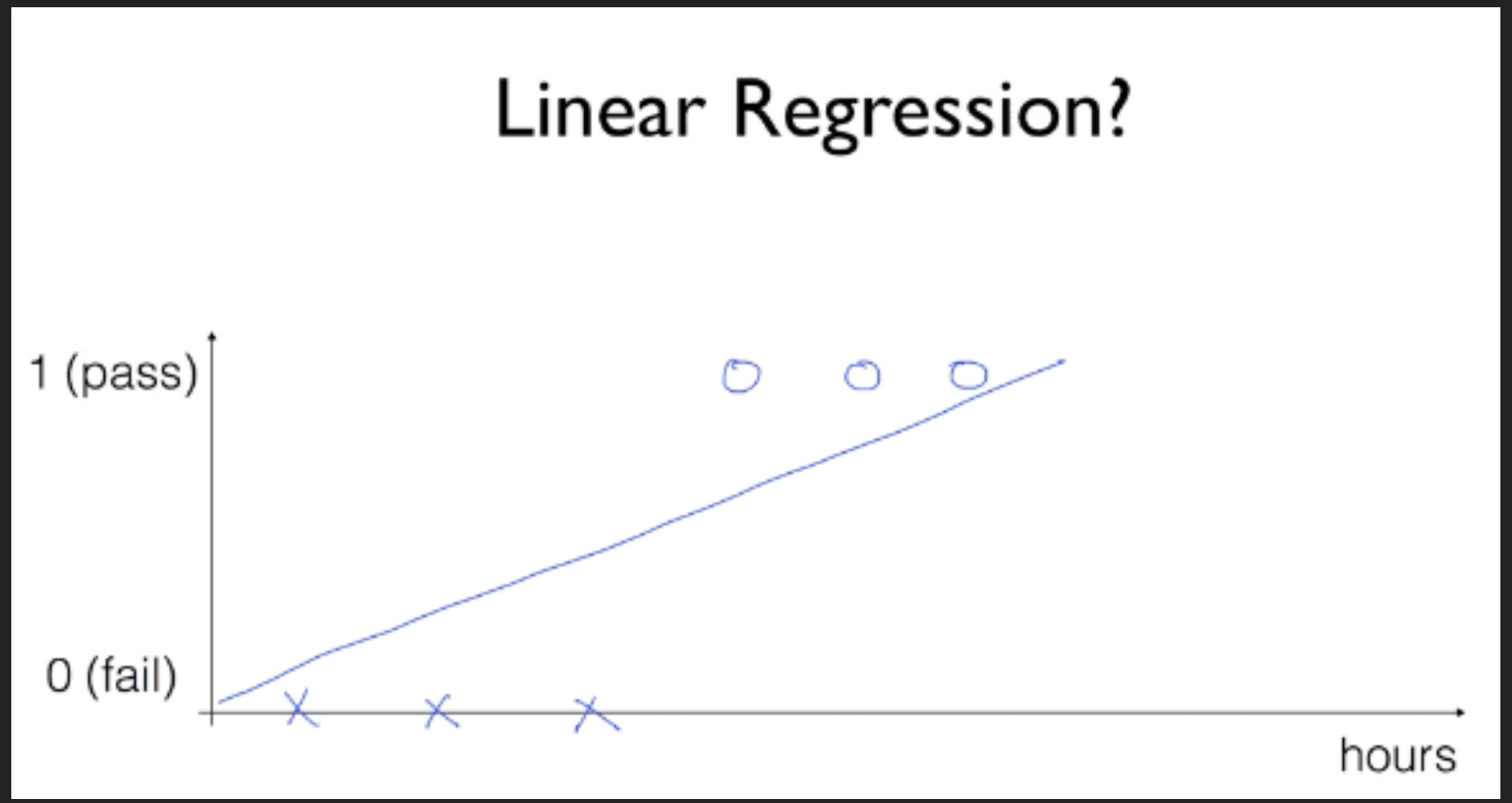
LOGISTIC CLASSIFICATION

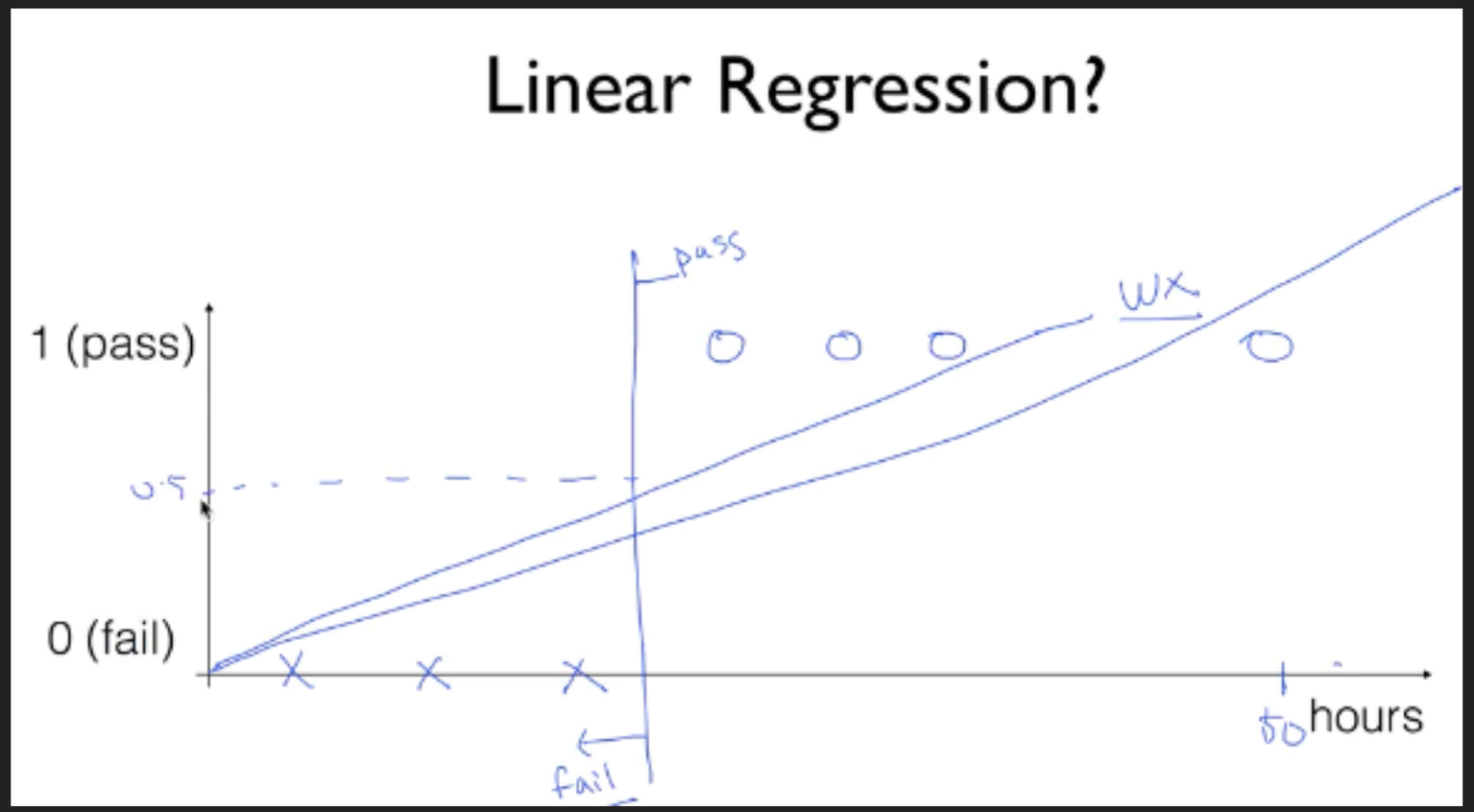
CLASSIFICATION AND ENCODING

- Spam Detection: Spam or Ham
- Facebook feed: show or hide
- Credit Card Fraudulent Transaction detection: legitimate/fraud

- Spam Detection: Spam (1) or Ham (0)
- Facebook feed: show(1) or hide(0)
- Credit Card Fraudulent Transaction detection: legitimate(0) or fraud (1)

LINEAR REGRESSION?





PROBLEMS

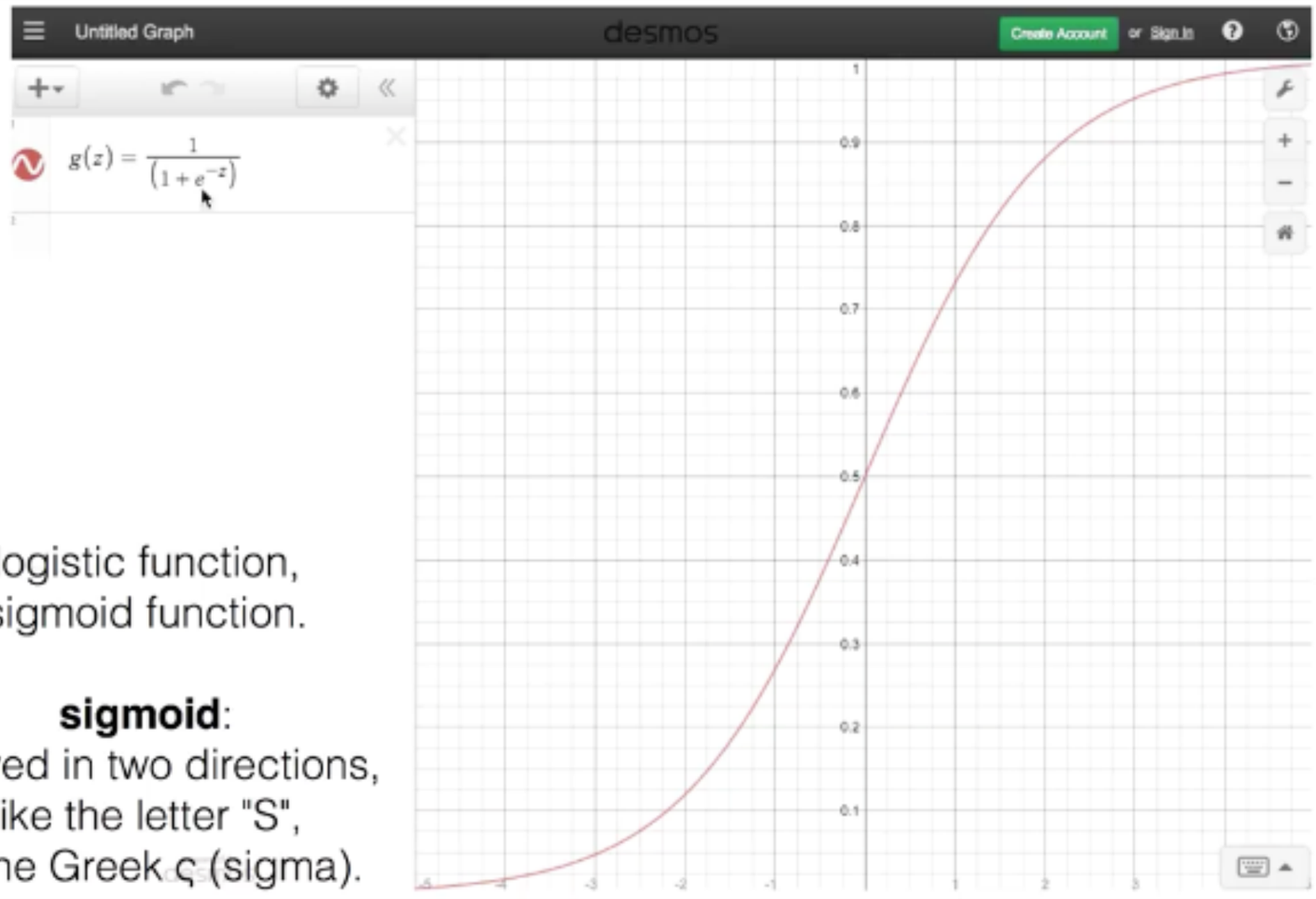
Linear regression

- We know Y is 0 or 1

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

- Hypothesis can give values large than 1 or less than 0

SIGMOID



logistic function,
sigmoid function.

sigmoid:
Curved in two directions,
like the letter "S",
or the Greek ς (sigma).

Logistic Hypothesis

$$H(X) = \frac{1}{1+e^{-W^T X}}$$

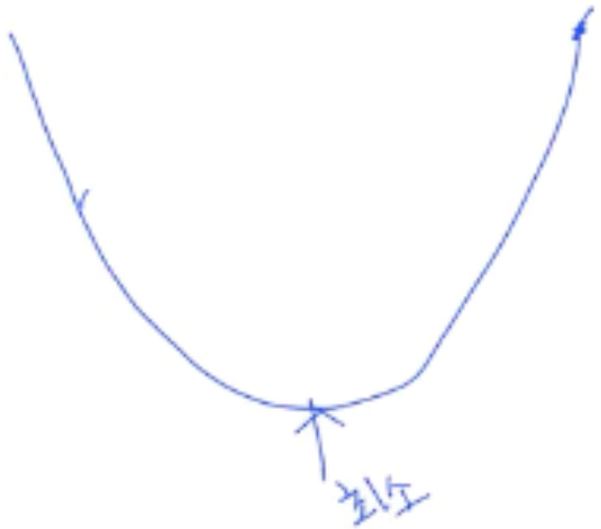
COST FUNCTION

Cost function

$$cost(W,b) = \frac{1}{m}\sum_{i=1}^{m}(H(x^{(i)}) - y^{(i)})^2$$

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

$$H(X) = \frac{1}{1+e^{-W^T X}}$$



COST FUNCTION FOR LOGISTIC

New cost function for logistic

$$cost(W) = \frac{1}{m}\sum \quad c(H(x), y)$$

$$c(H(x), y) = \begin{cases} -log(H(x)) & : y = 1 \\ -log(1 - H(x)) & : y = 0 \end{cases}$$

COST FUNCTION IS LOG FUNCTION

understanding cost function

$$C(H(x), y) = \begin{cases} -log(H(x)) & : y = 1 \\ -log(1 - H(x)) & : y = 0 \end{cases}$$

1 + e^-2

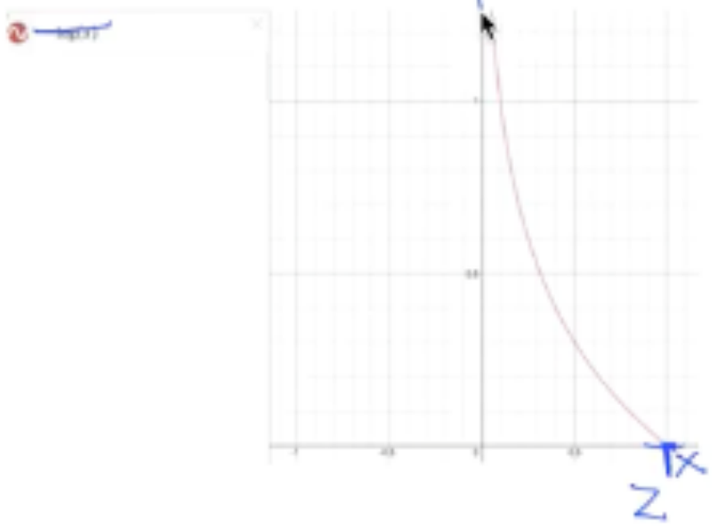
log

Cost y=1

H(x) = 1 → cost(1) = 0

H(x) = 0 → cost = ∞

cost = -log(z)



y=0

H(x) = 0, cost = 0

H(x) = 1, cost = ∞

-log(1-z)

COST FUNCTION

Cost function

$$cost(W) = \frac{1}{m}\sum \quad c\,(H(x), y)$$

$$c\,(H(x), y) = \begin{cases} -log(H(x)) & : y = 1 \\ -log(1 - H(x)) & : y = 0 \end{cases}$$

$$c\,(H(x), y) = -ylog(H(x)) - (1 - y)log(1 - H(x))$$

GRADIENT DESCENT ALGORITHM

Gradient decent algorithm

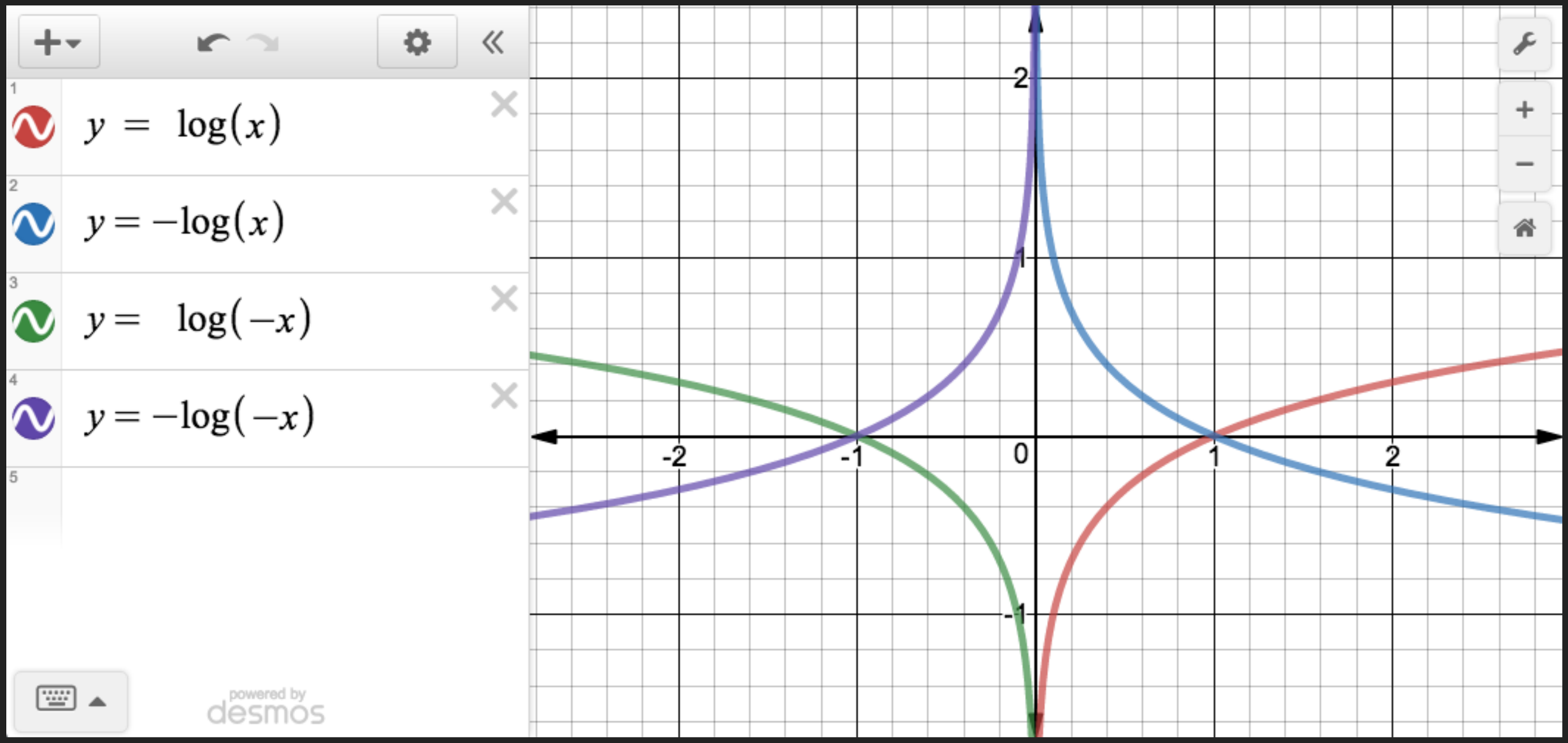
$$cost(W) = -\frac{1}{m}\sum ylog(H(x)) + (1-y)log(1-H(x))$$

$$W := W - \alpha\frac{\partial}{\partial W}cost(W)$$

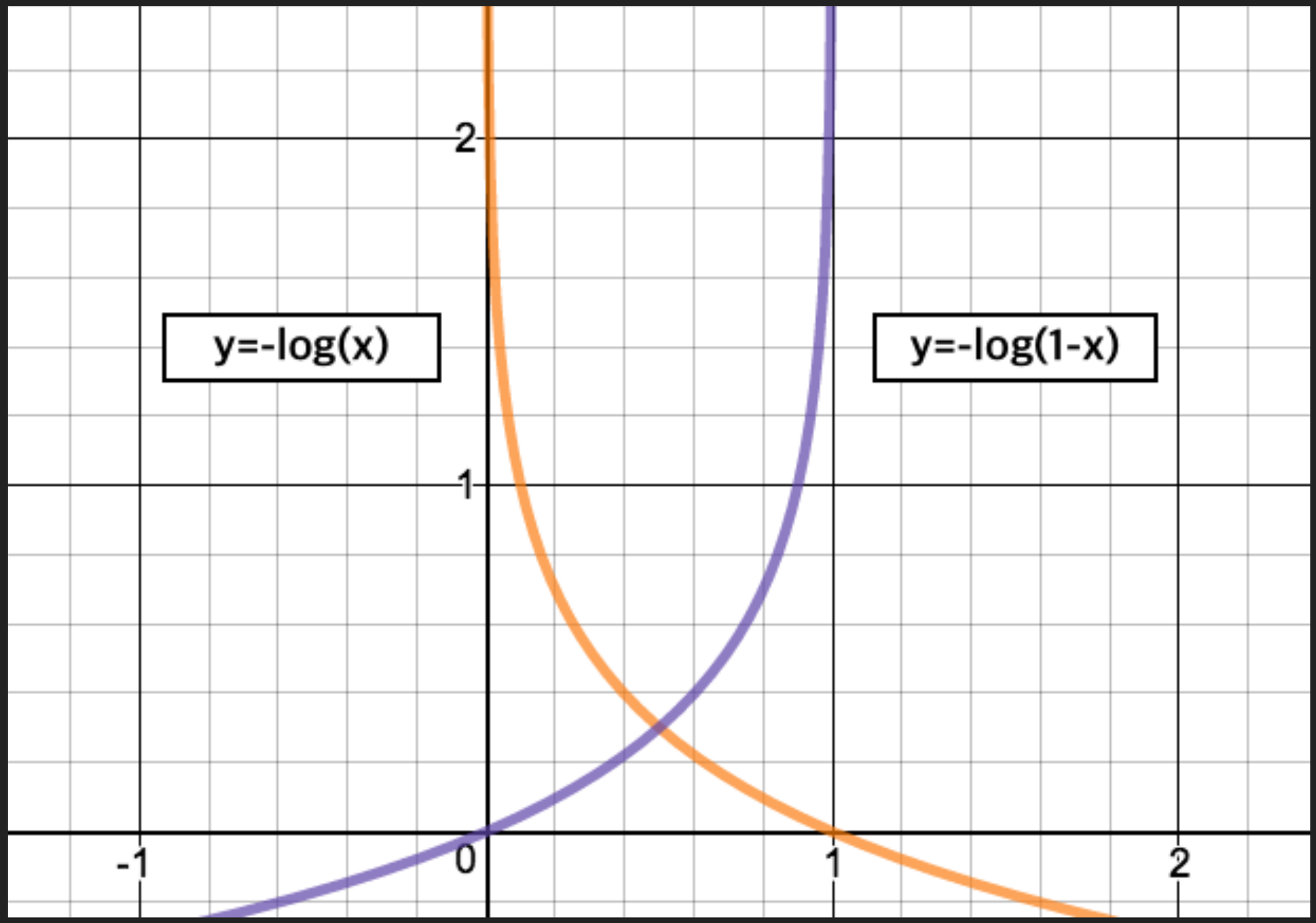
```
# cost function
cost = tf.reduce_mean(-tf.reduce_sum(Y*tf.log(hypothesis) + (1-Y)*tf.log(1-hypothesis)))

# Minimize
a = tf.Variable(0.1) # Learning rate, alpha
optimizer = tf.train.GradientDescentOptimizer(a)
train = optimizer.minimize(cost)
```

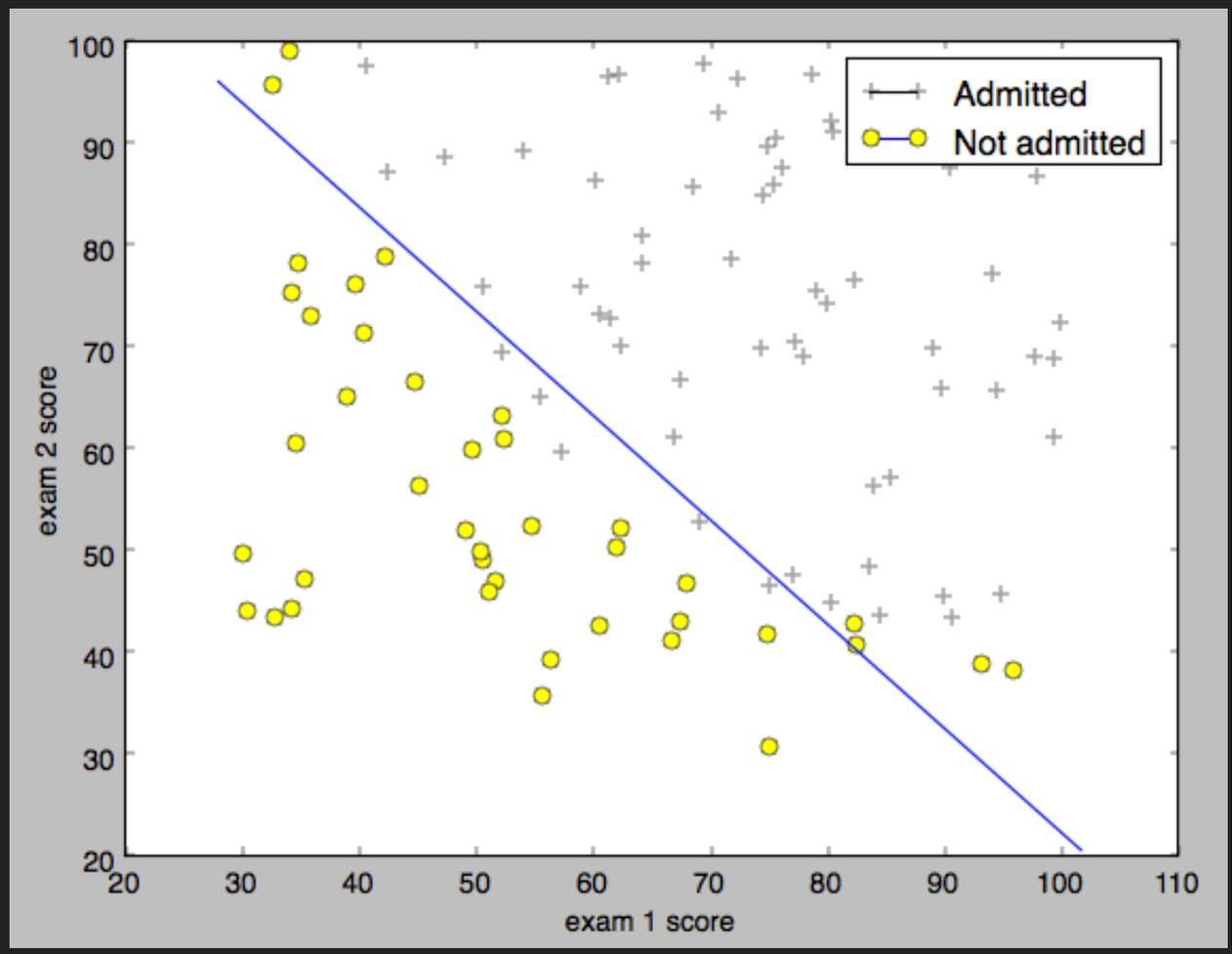
LOG FUNCTION

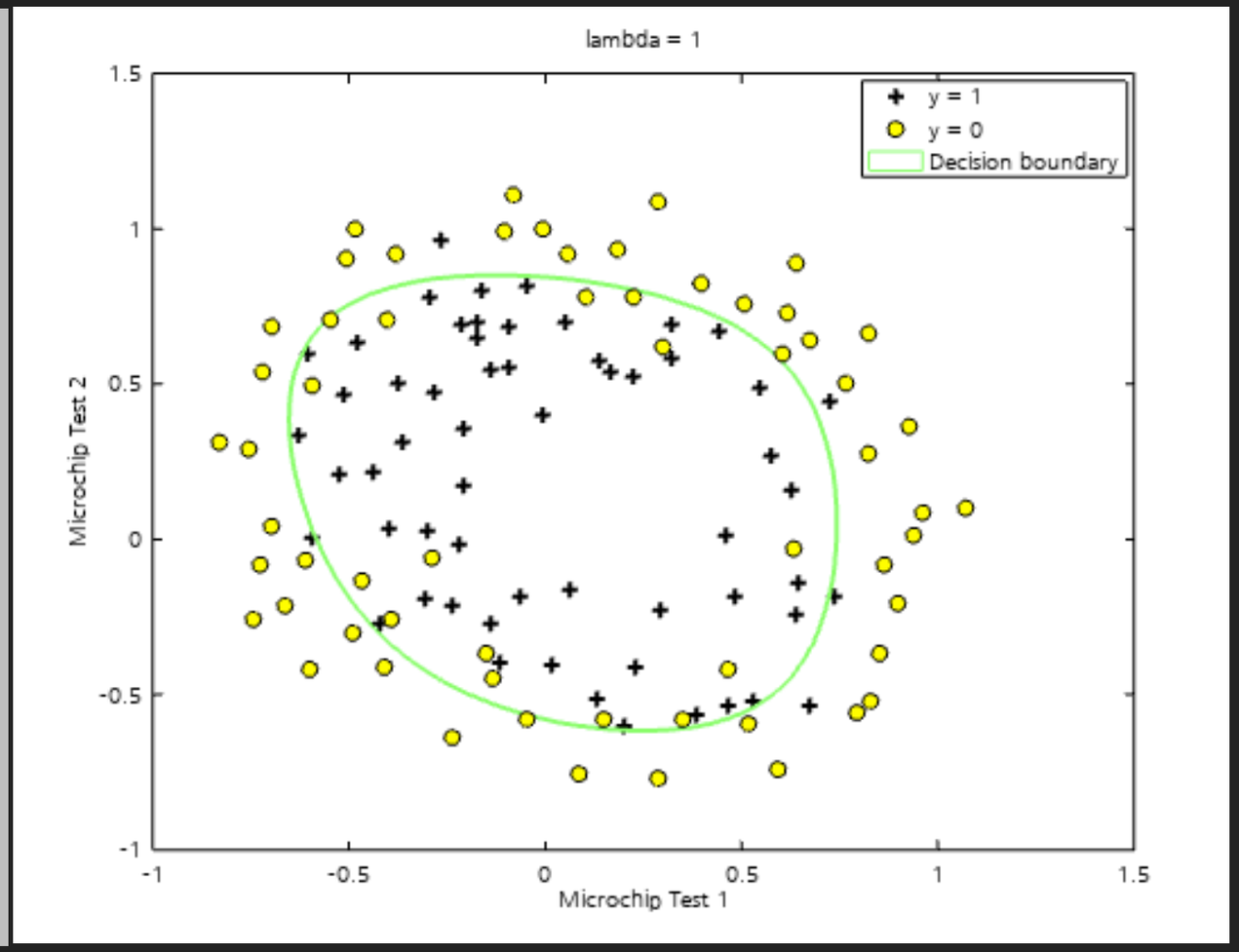


LOG FUNCTIONS WE NEED TO KNOW



DECISION BOUNDARY

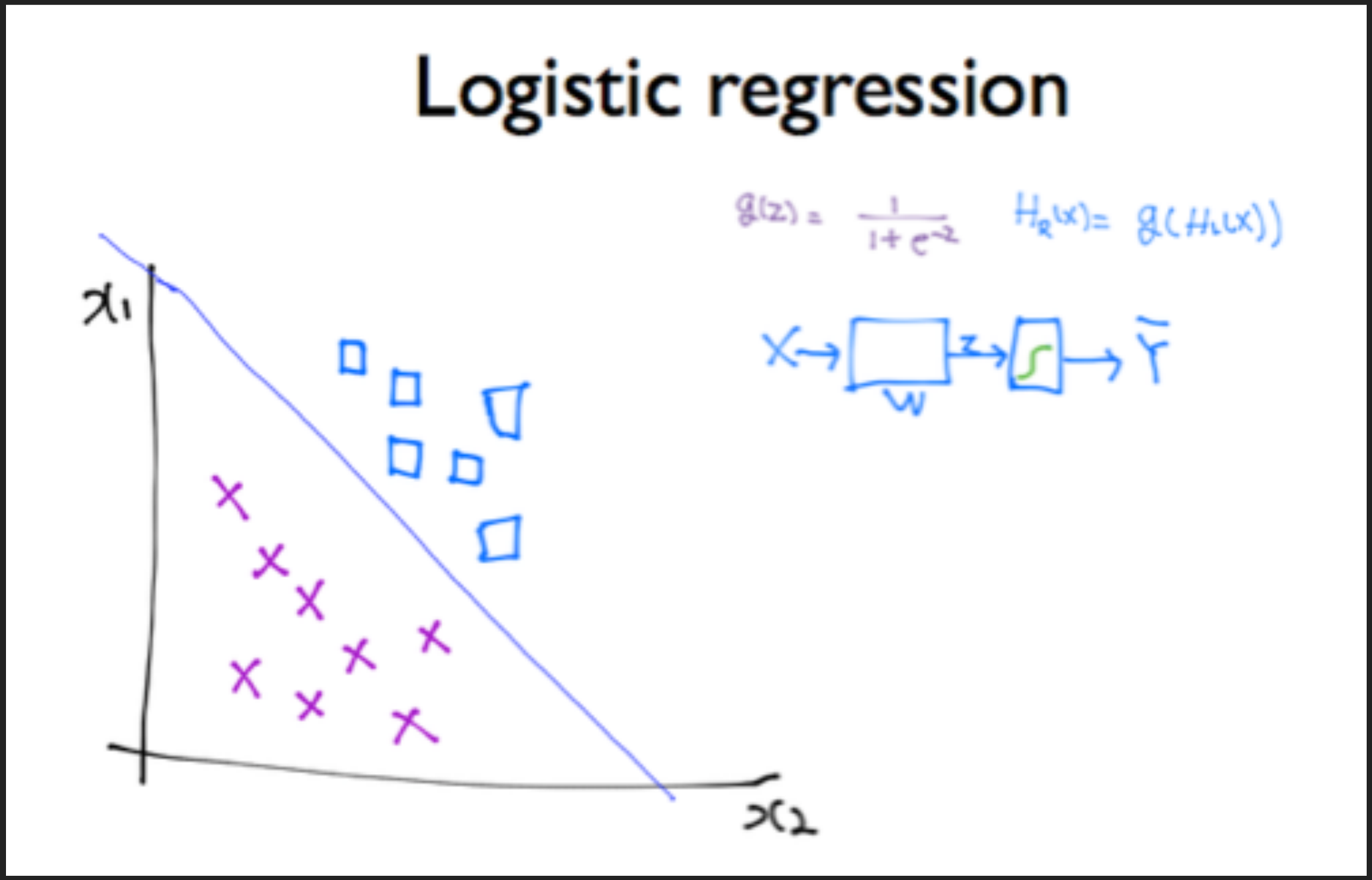




MACHINE LEARNING

SOFTMAX

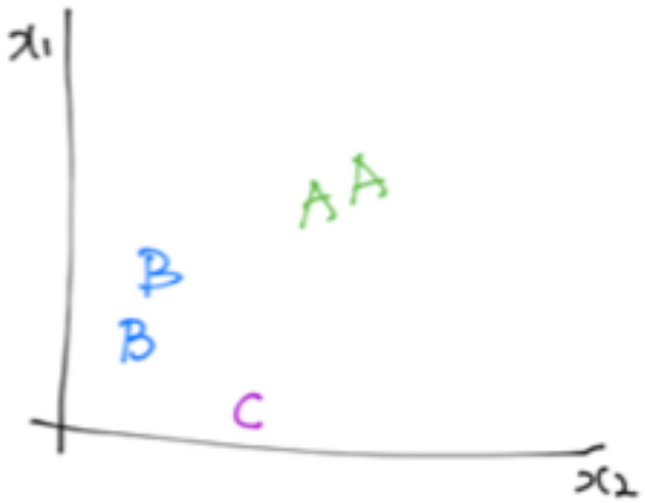
LOGISTIC REGRESSION



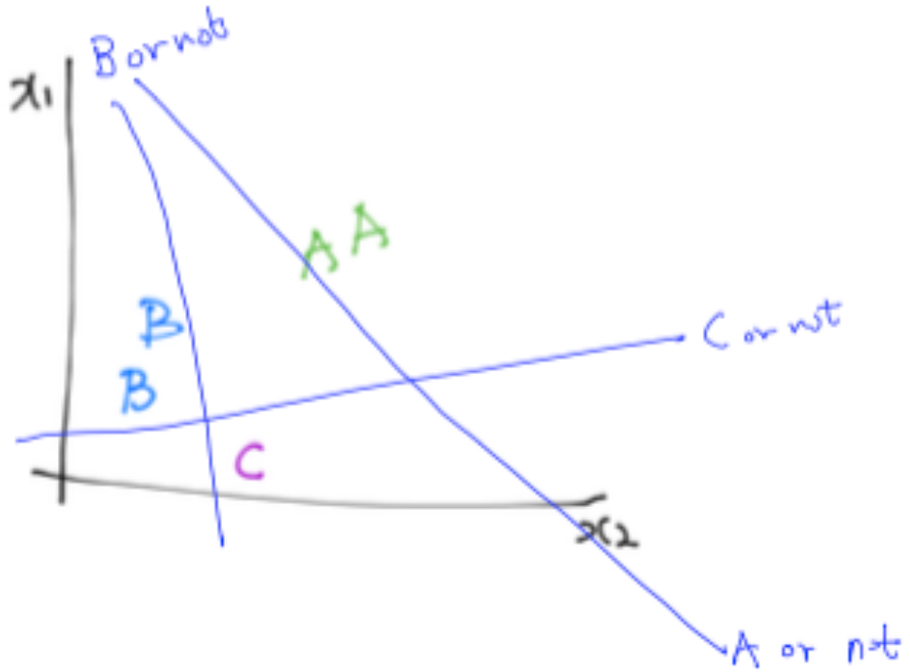
MULTINOMIAL CLASSIFICATION

Multinomial classification

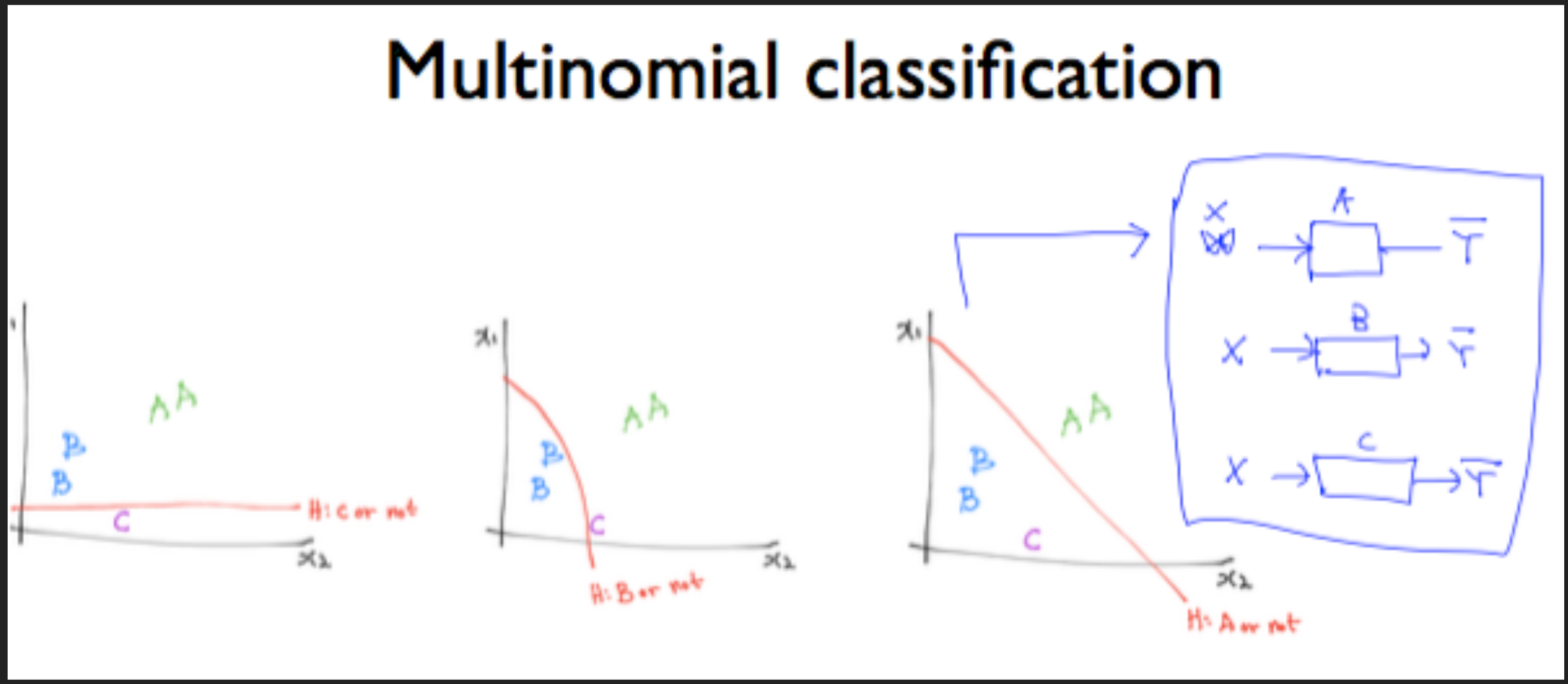
x1 (hours)	x2 (attendance)	y (grade)
10	5	A
9	5	A
3	2	B
2	4	B
11	1	C



Multinomial classification



MULTINOMIAL CLASSIFICATION



MULTINOMIAL CLASSIFICATION

$$[w_1 \quad w_2 \quad w_3]\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = [w_1x_1 + w_2x_2 + w_3x_3]$$

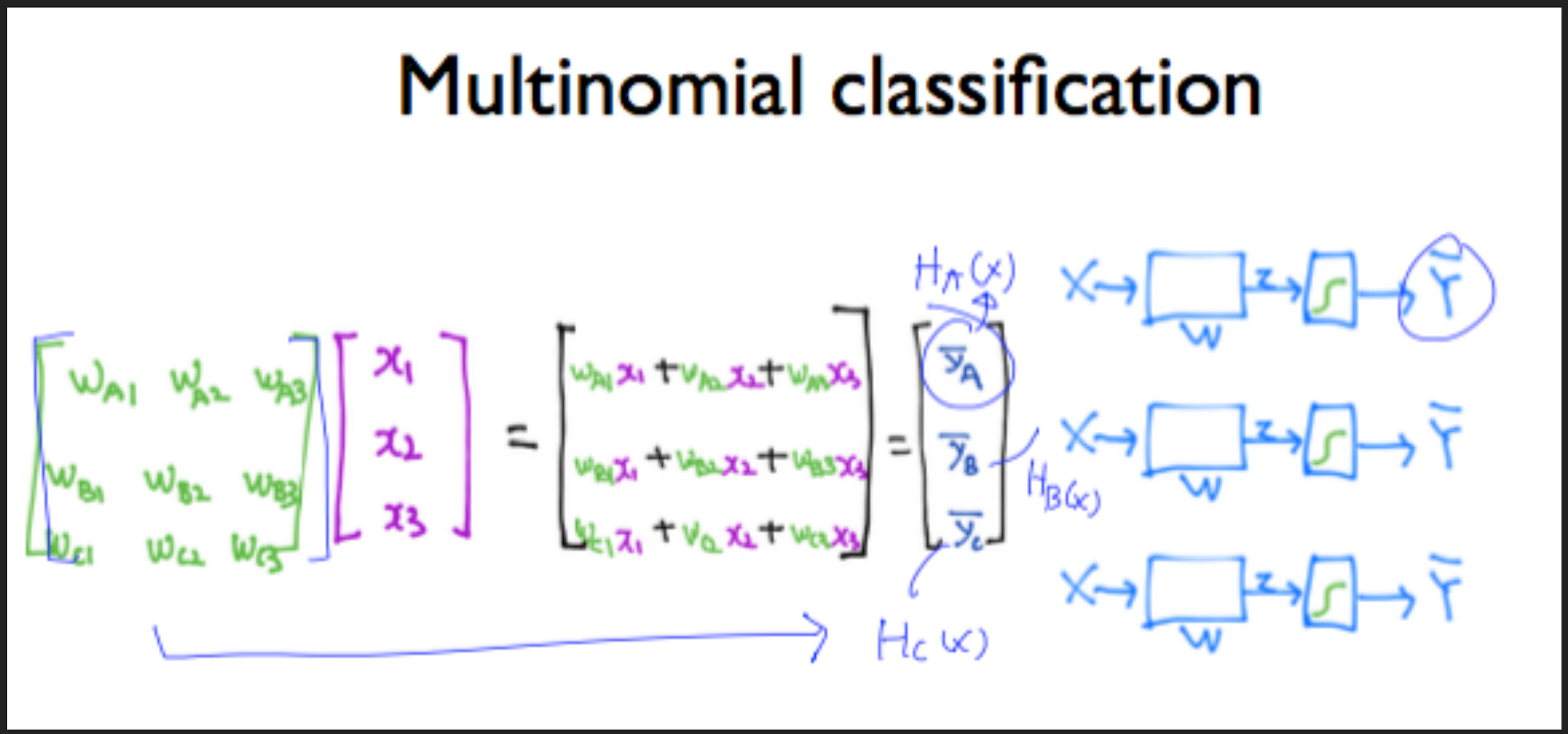
$$[w_1 \quad w_2 \quad w_3]\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = [w_1x_1 + w_2x_2 + w_3x_3]$$

$$[w_1 \quad w_2 \quad w_3]\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = [w_1x_1 + w_2x_2 + w_3x_3]$$

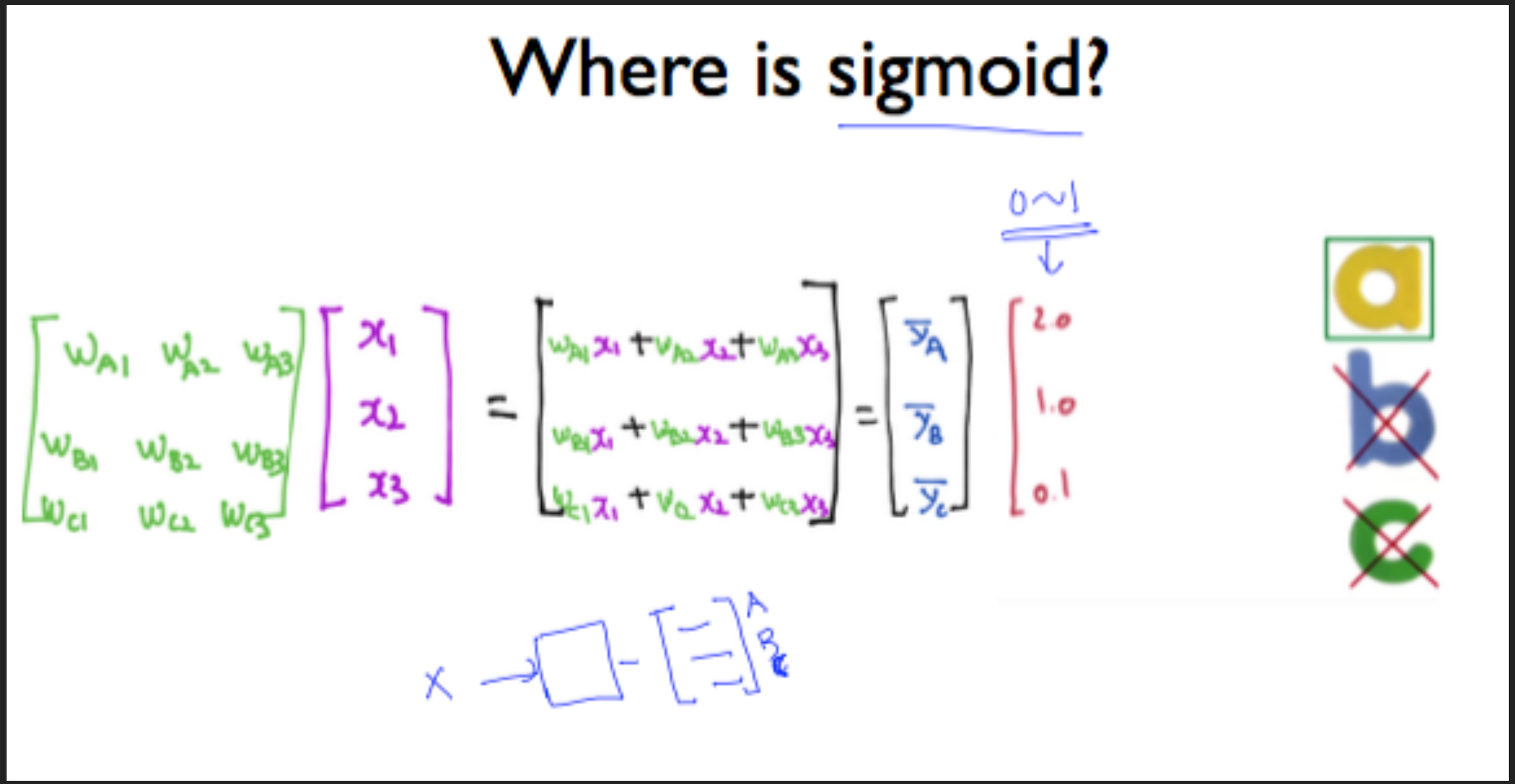
$$[w_1 \quad w_2 \quad w_3]\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = [w_1x_1 + w_2x_2 + w_3x_3]$$

$$\begin{bmatrix} w_{A1} & w_{A2} & w_{A3} \\ w_{B1} & w_{B2} & w_{B3} \\ w_{C1} & w_{C2} & w_{C3} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} =$$

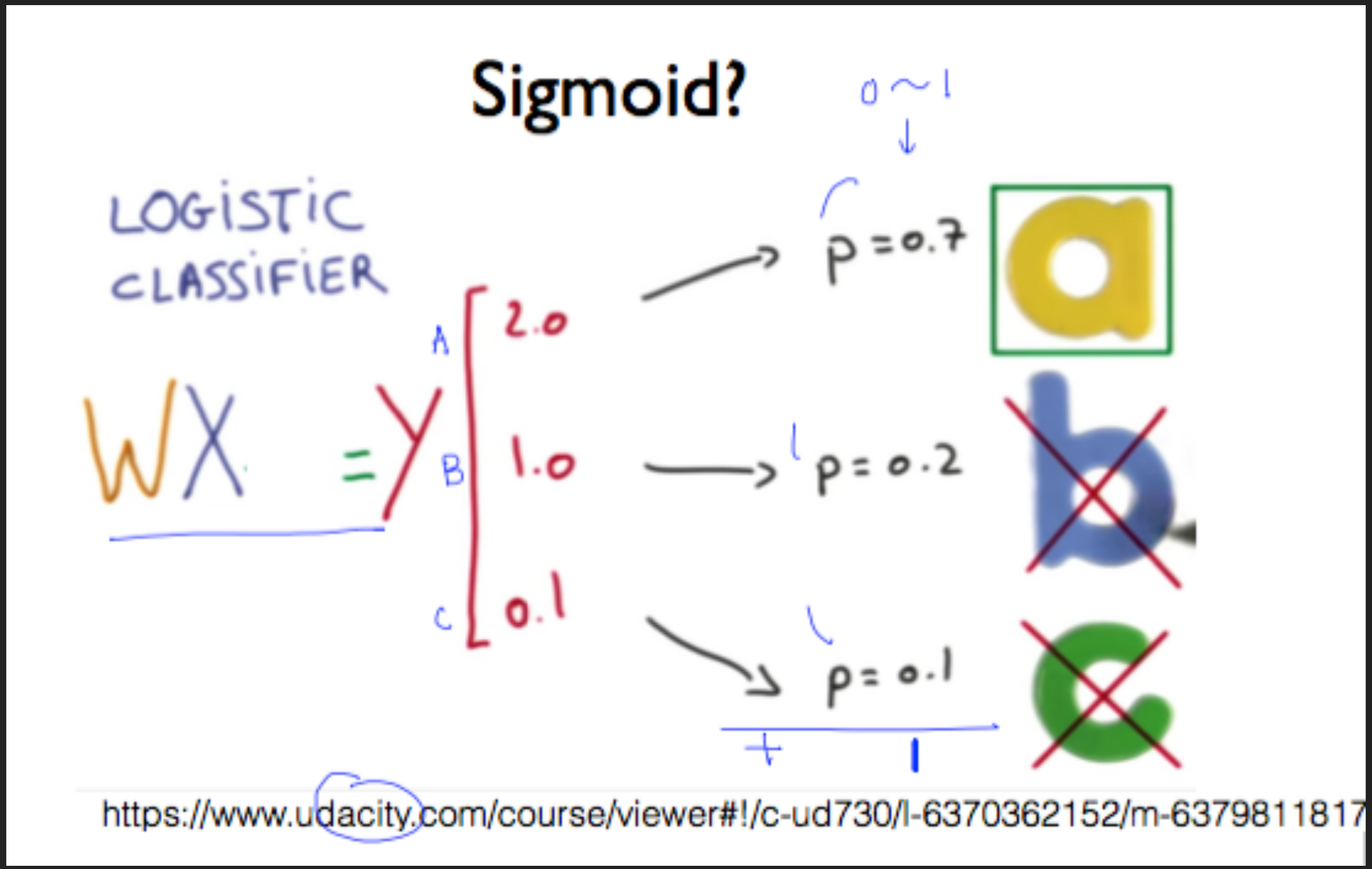
MULTINOMIAL CLASSIFICATION



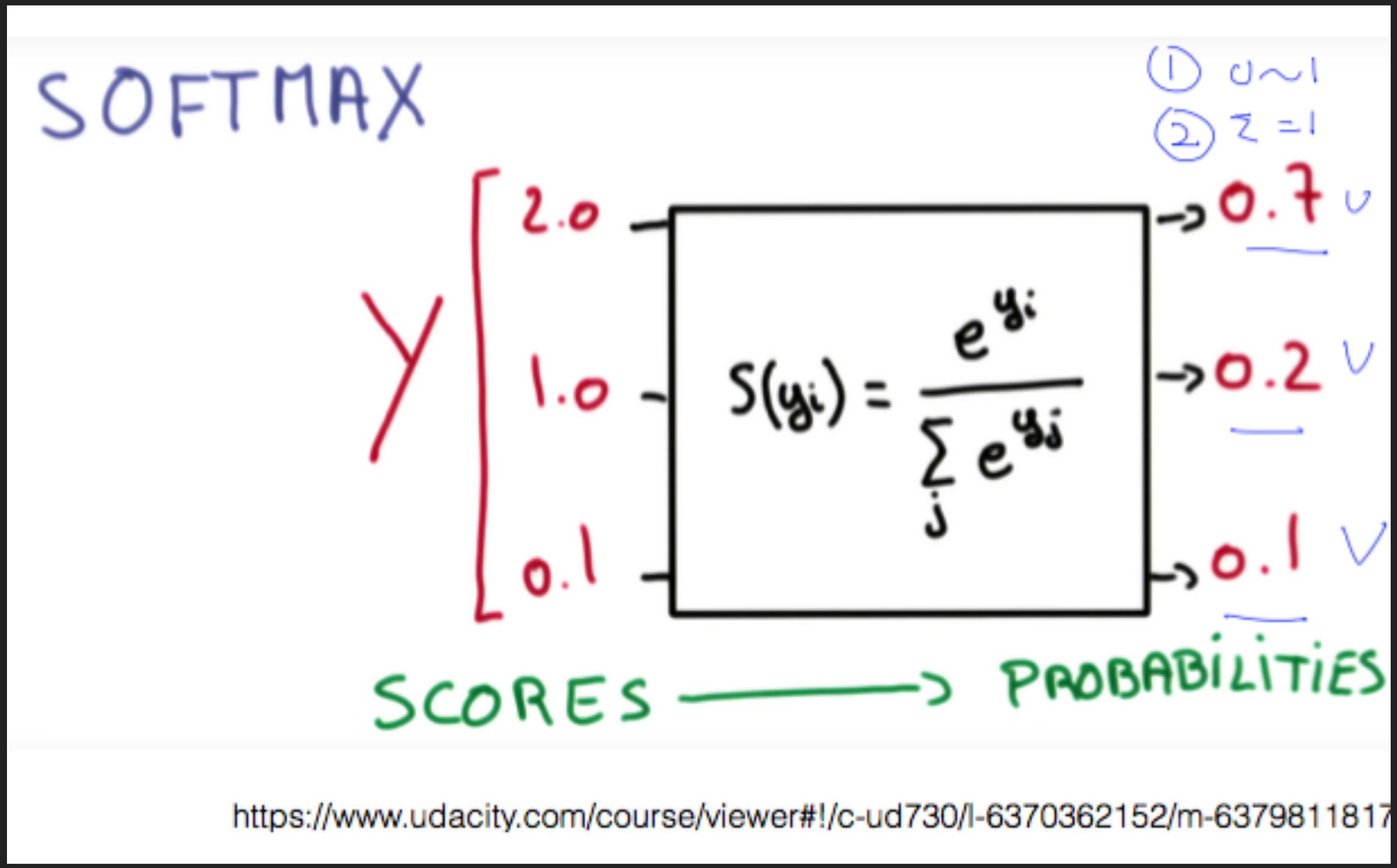
WHERE IS SIGMOID?



SIGMOID?


Sigmoid?
0 ~ 1
LOGISTIC CLASSIFIER
WX = Y
A 2.0
B 1.0
C 0.1
p = 0.7
p = 0.2
p = 0.1
+
1
https://www.udacity.com/course/viewer#!/c-ud730/l-6370362152/m-6379811817

SOFTMAX

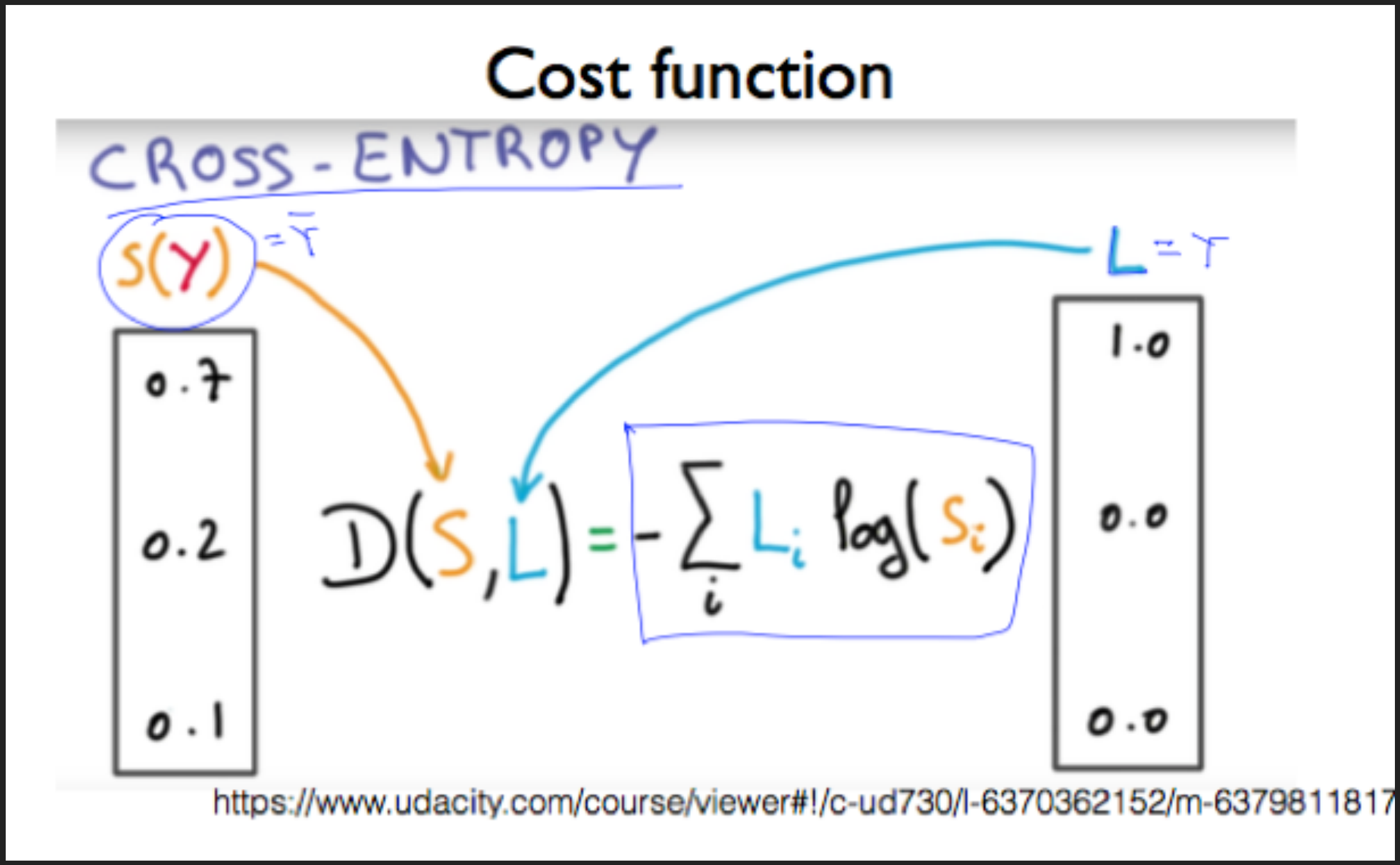


https://www.udacity.com/course/viewer#!/c-ud730/l-6370362152/m-6379811817

ONE-HOT ENCODING

https://www.udacity.com/course/viewer#!/c-ud730/l-6370362152/m-6379811817

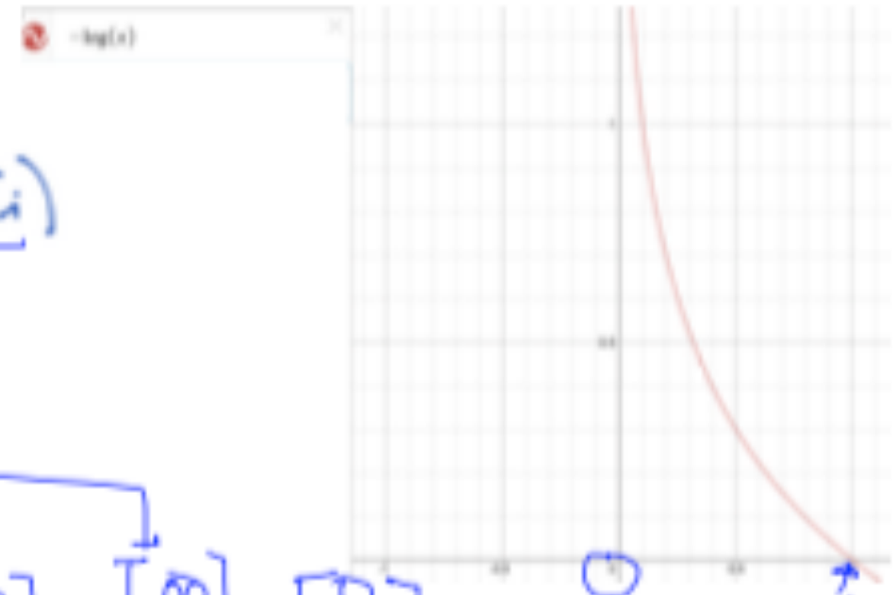
COST FUNCTION



CROSS-ENTROPY COST FUNCTION

Cross-entropy cost function

$$-\sum_i L_i \log(S_i) \qquad -\sum_i L_i \log(\bar{y}_i) = \sum_i L_i \times -\log(\bar{y}_i)$$



$Y = L = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = B$

$\hat{Y} = A \begin{bmatrix} 0 \\ 1 \end{bmatrix} B \quad (OK) \quad \begin{bmatrix} 0 \\ 1 \end{bmatrix} \odot -\log \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \odot \begin{bmatrix} \infty \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow 0$

$\hat{Y} = A \begin{bmatrix} 1 \\ 0 \end{bmatrix} B \quad (X) \quad \begin{bmatrix} 0 \\ 1 \end{bmatrix} \odot -\log \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \odot \begin{bmatrix} 0 \\ \infty \end{bmatrix} = \begin{bmatrix} 0 \\ \infty \end{bmatrix} \Rightarrow \infty$

전개

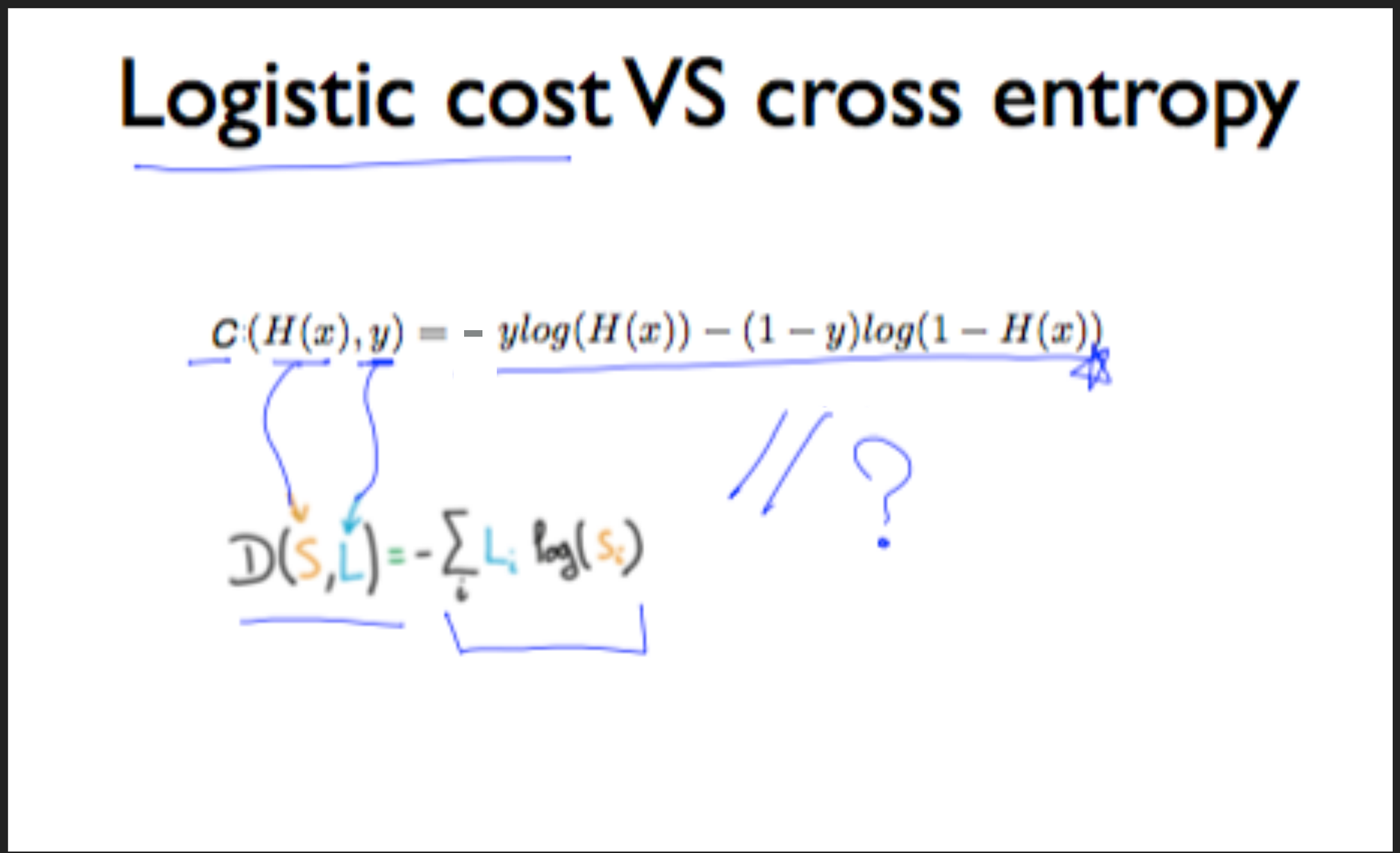
$$-\sum_i L_i \log(S_i) \quad => \quad -\sum_i L_i \log(\hat{y}_i) \quad => \quad \sum_i L_i * \ -\log(\hat{y}_i)$$

$$L = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

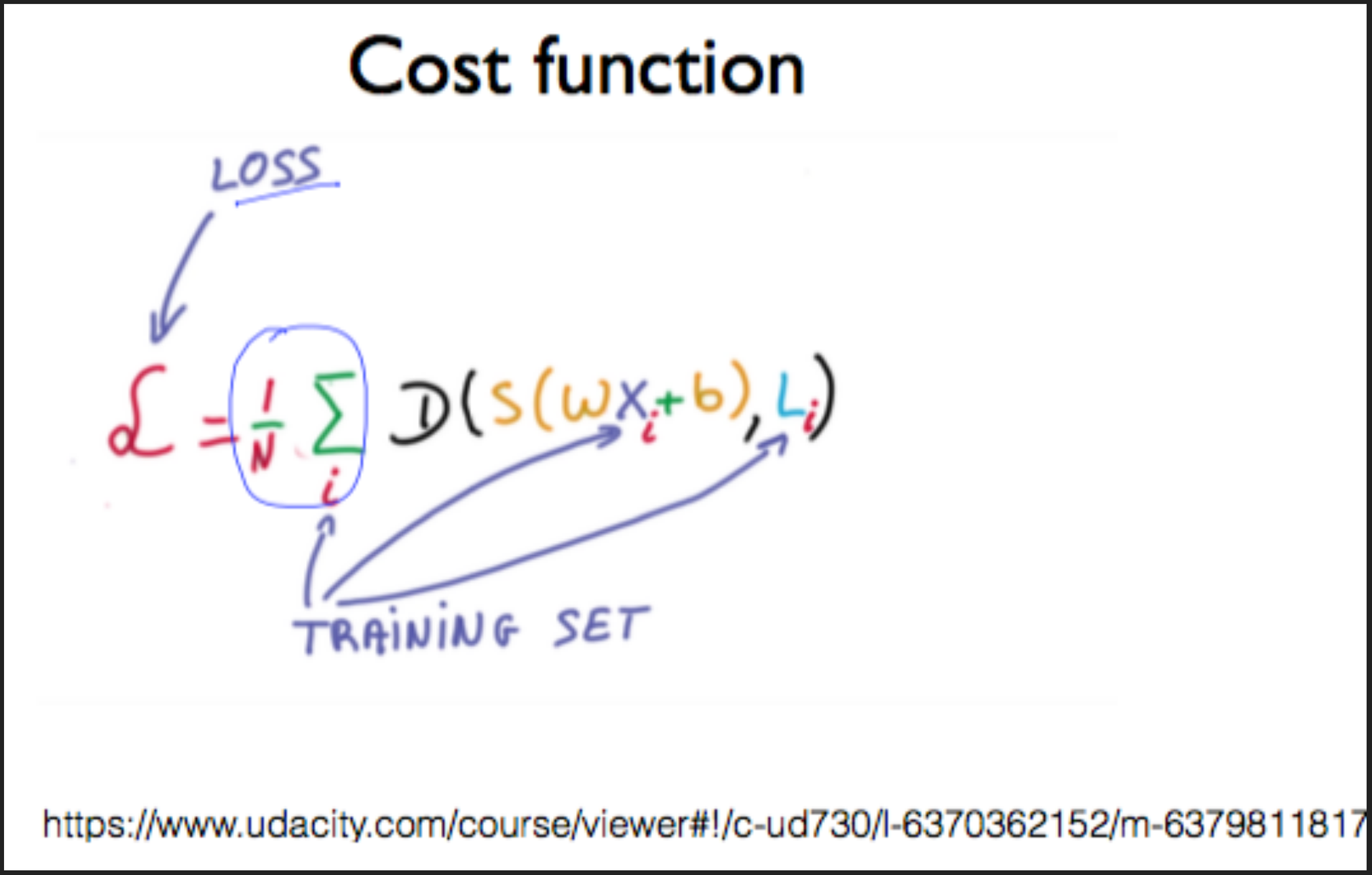
$$\hat{Y} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 1 \end{bmatrix} * -log \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} * \begin{bmatrix} \infty \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = 0, \quad cost=0$$

$$\hat{Y} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 1 \end{bmatrix} * -log \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} * \begin{bmatrix} 0 \\ \infty \end{bmatrix} = \begin{bmatrix} 0 \\ \infty \end{bmatrix} = \infty, \quad cost=\infty$$

LOGISTIC COST VS. CROSS ENTROPY

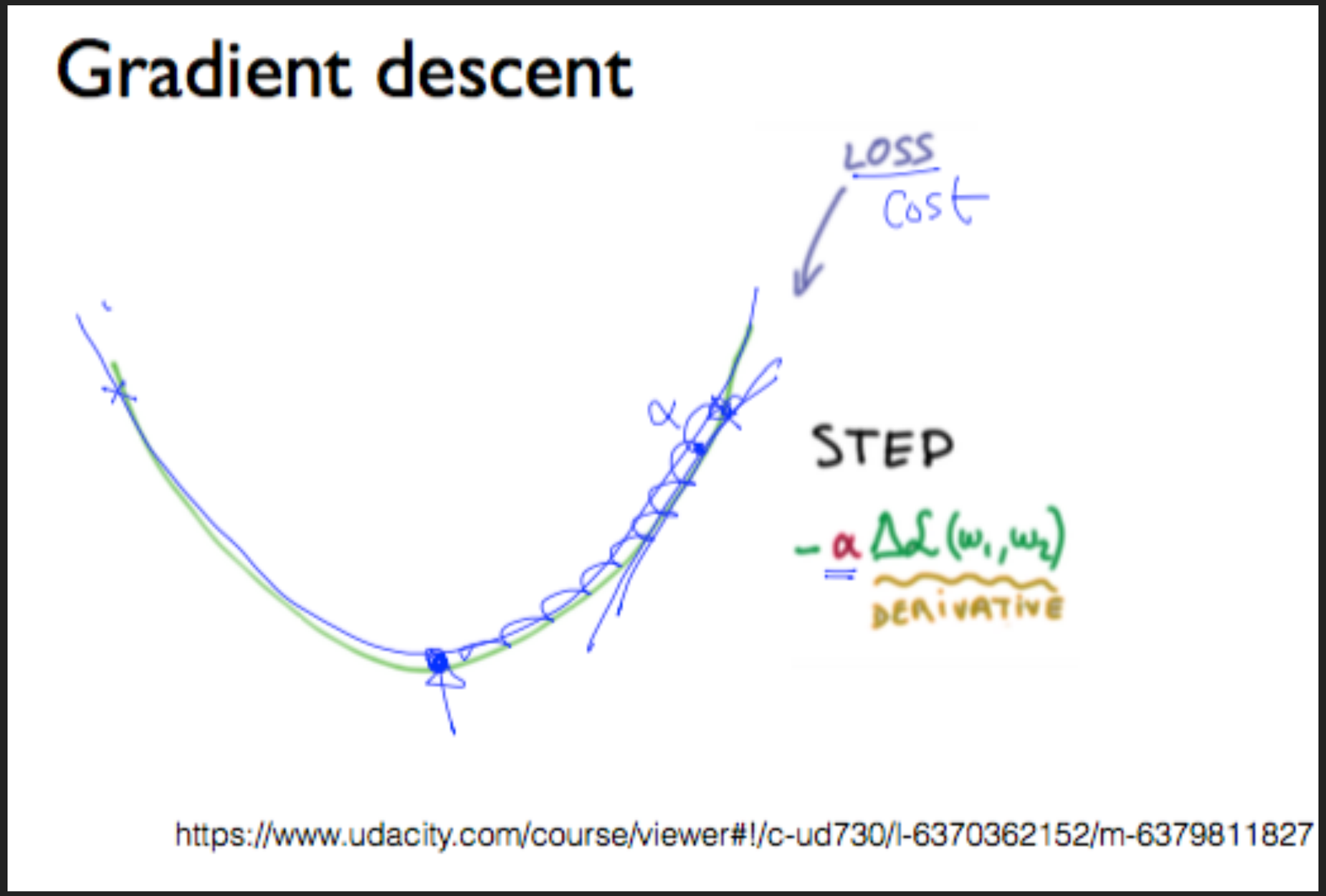


COST FUNCTION



https://www.udacity.com/course/viewer#!/c-ud730/l-6370362152/m-6379811817

GRANDIENT DESCENT ALGORITHM



HOMEWORK

Logistic cost VS cross entropy

$$C(H(x), y) = -ylog(H(x)) - (1-y)log(1-H(x))$$

$$D(S, L) = -\sum_i L_i \log(S_i)$$

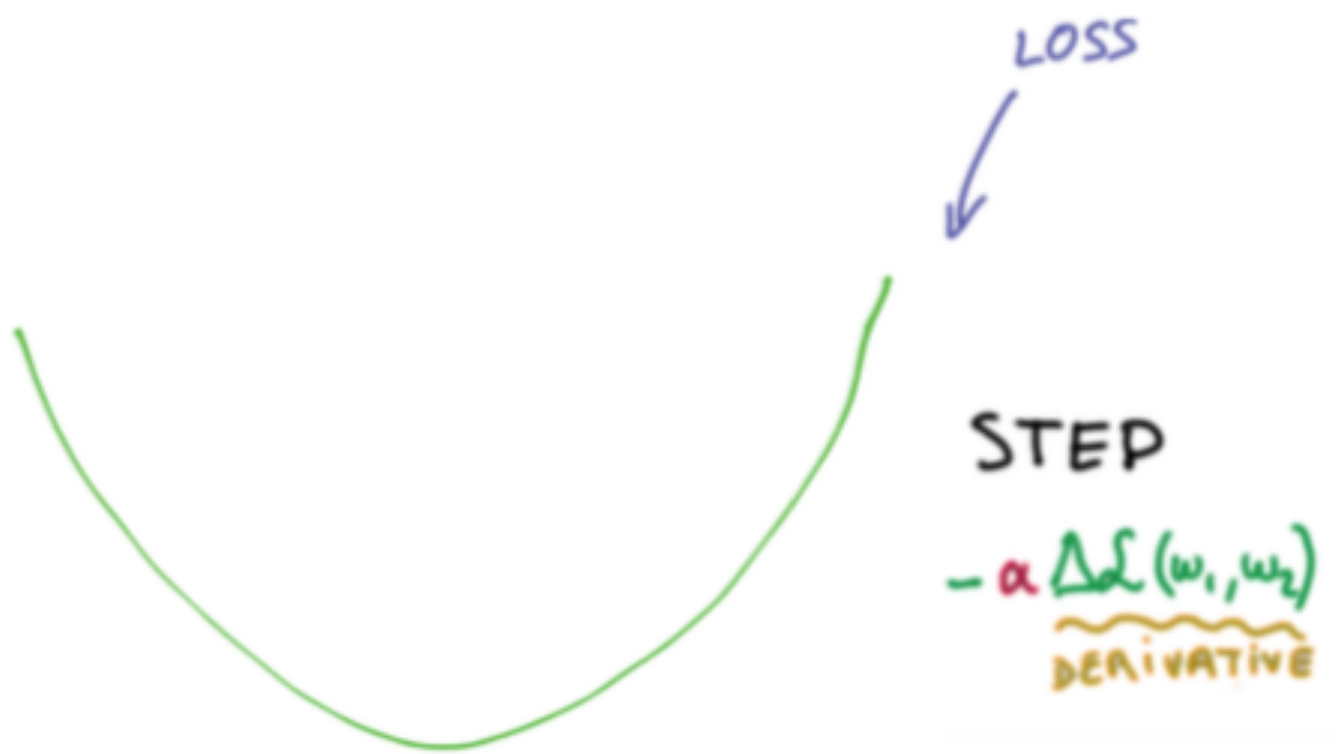
//?

MACHINE LEARNING

APPLICATION AND TIPS

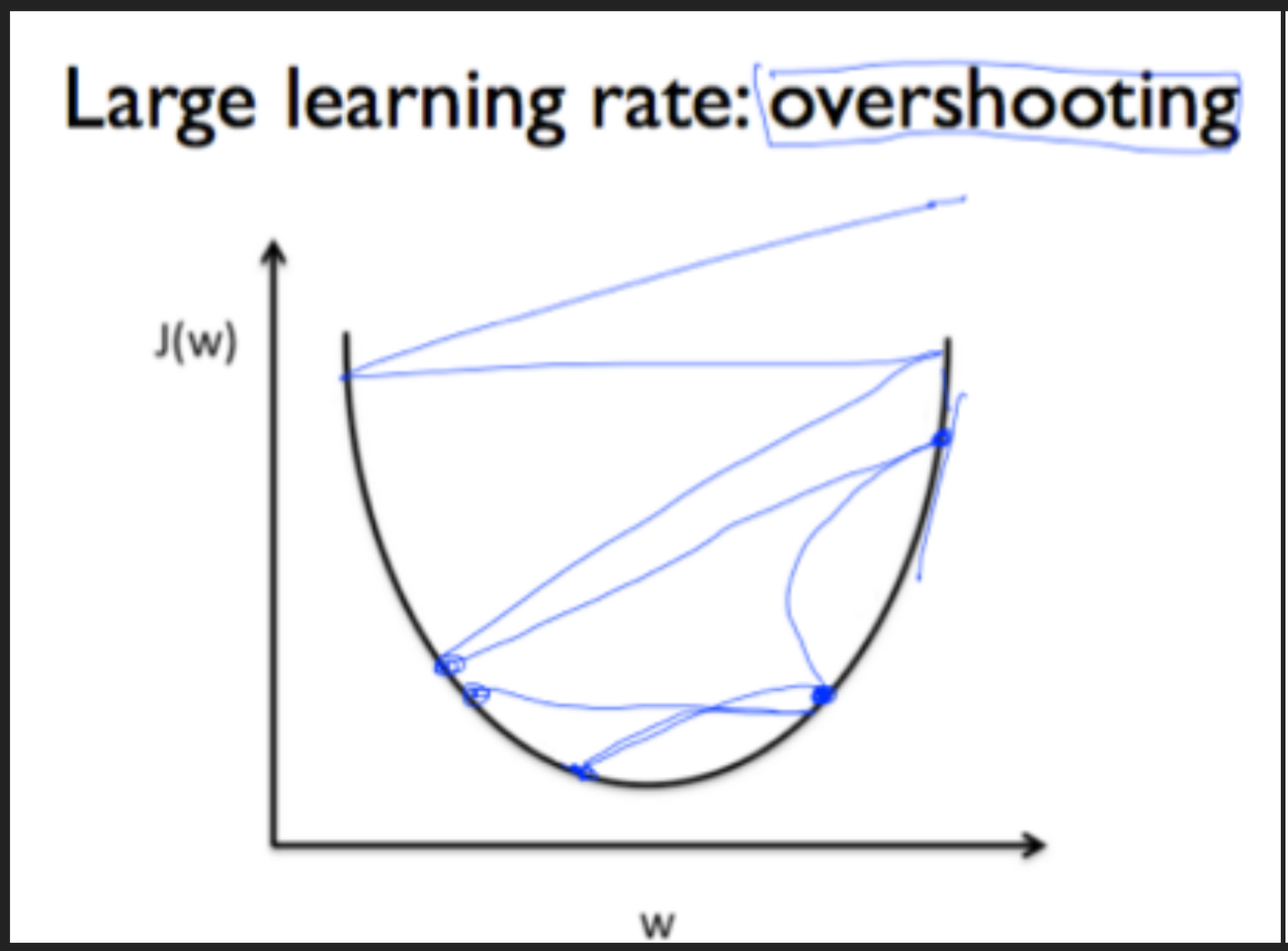
COST FUNCTION GRAPH

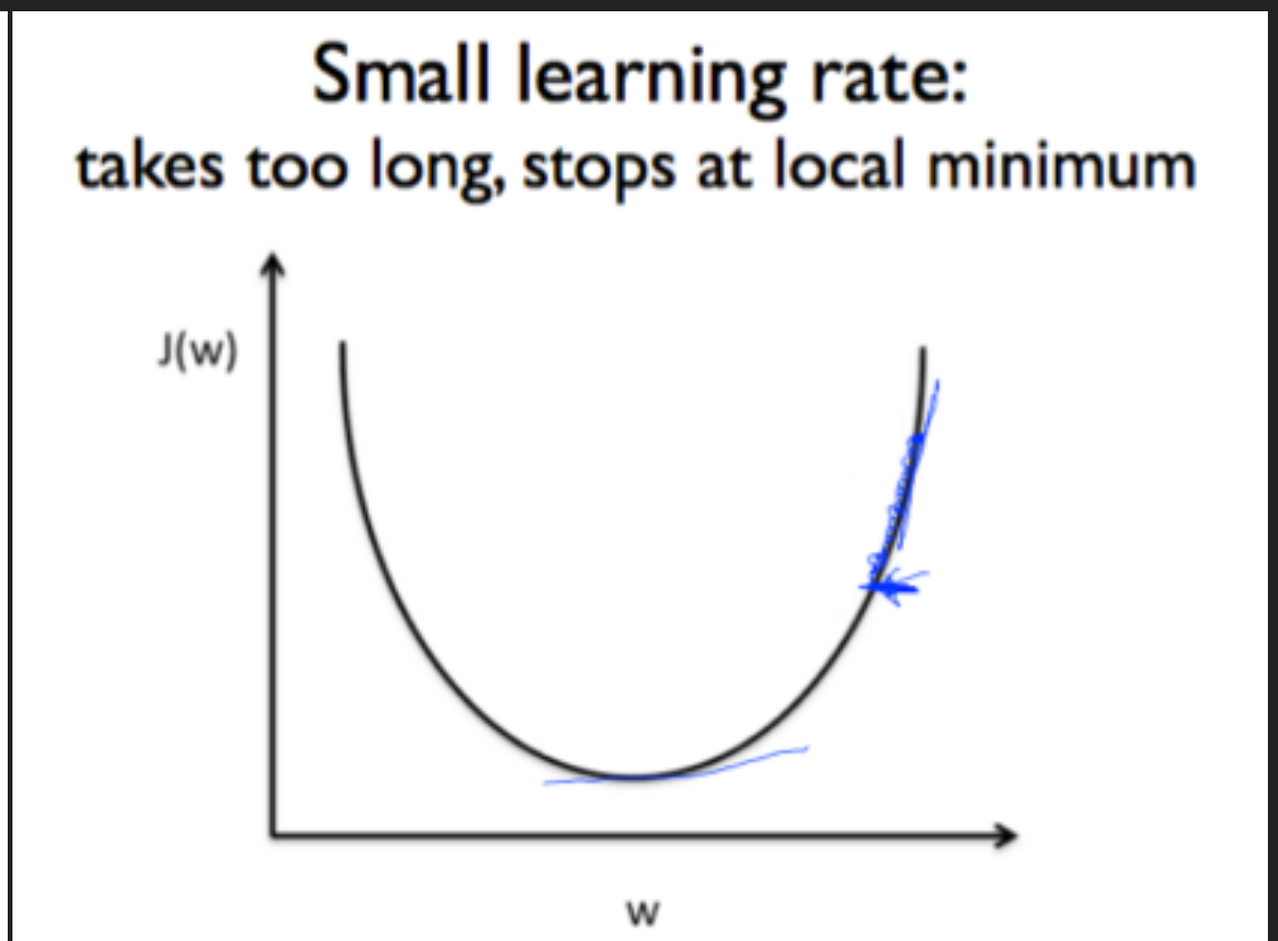
```
# Minimize error using cross entropy
learning_rate = 0.001
cost = tf.reduce_mean(-tf.reduce_sum(Y*tf.log(hypothesis), reduction_indices=1)) # Cross entropy
optimizer = tf.train.GradientDescentOptimizer(learning_rate).minimize(cost) # Gradient Descent
```



https://www.udacity.com/course/viewer#!/c-ud730/l-6370362152/m-6379811827

OVERSHOOTING AND SMALL LEARNING RATE





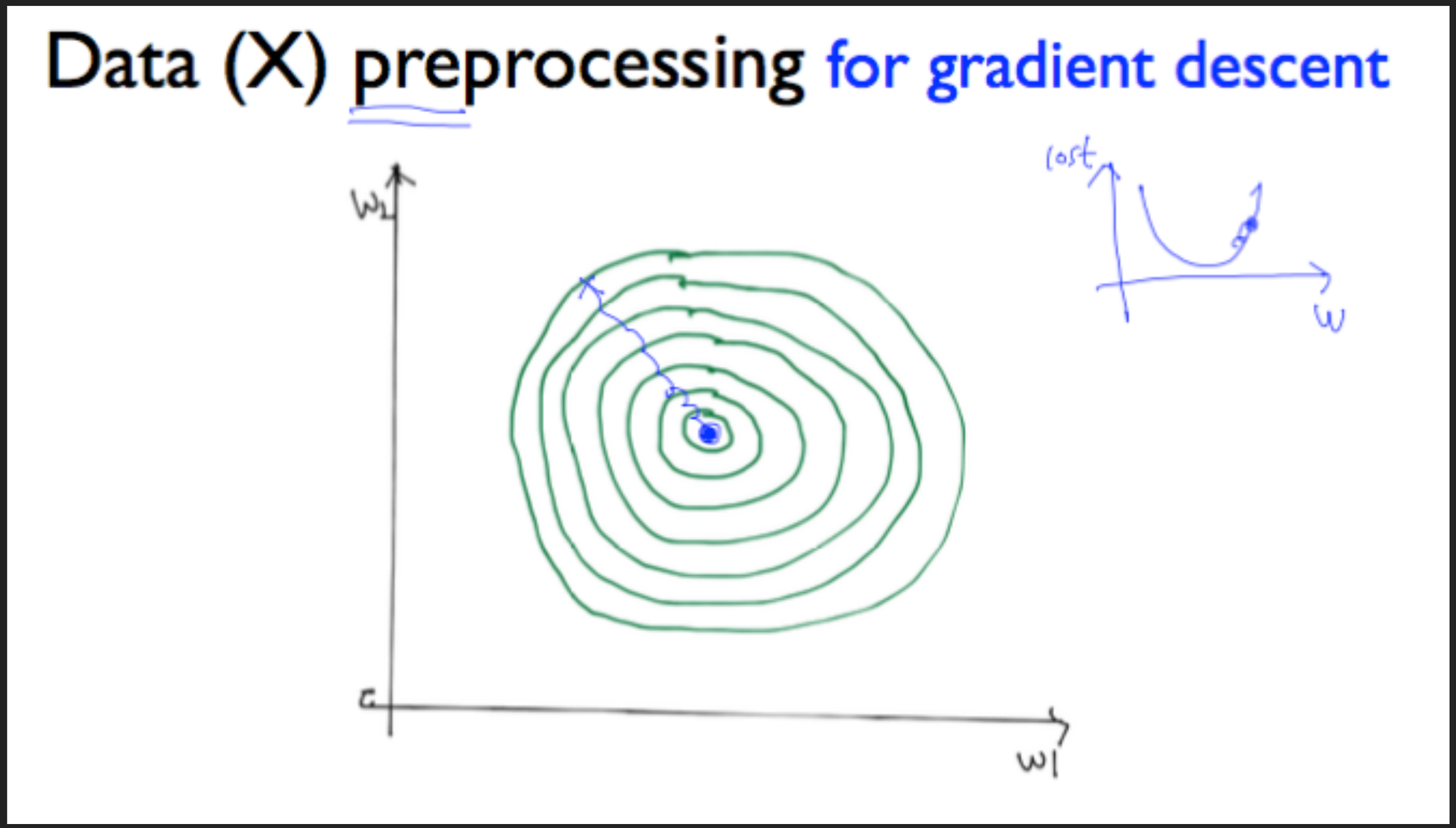
TRY SEVERAL LEARNING RATES

Try several learning rates

0.01

- Observe the cost function
- Check it goes down in a reasonable rate

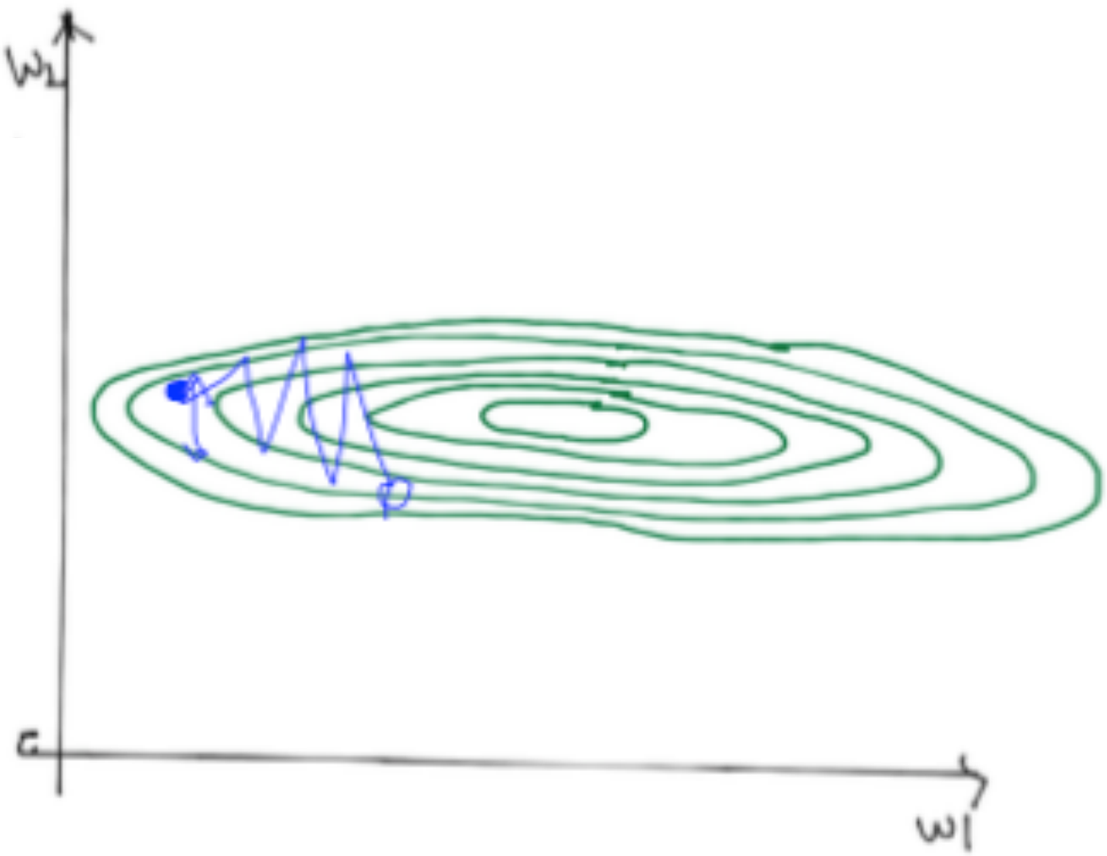
DATA PREPROCESSING FOR GRADIENT DESCENT



DATA PREPROCESSING FOR GRADIENT DESCENT

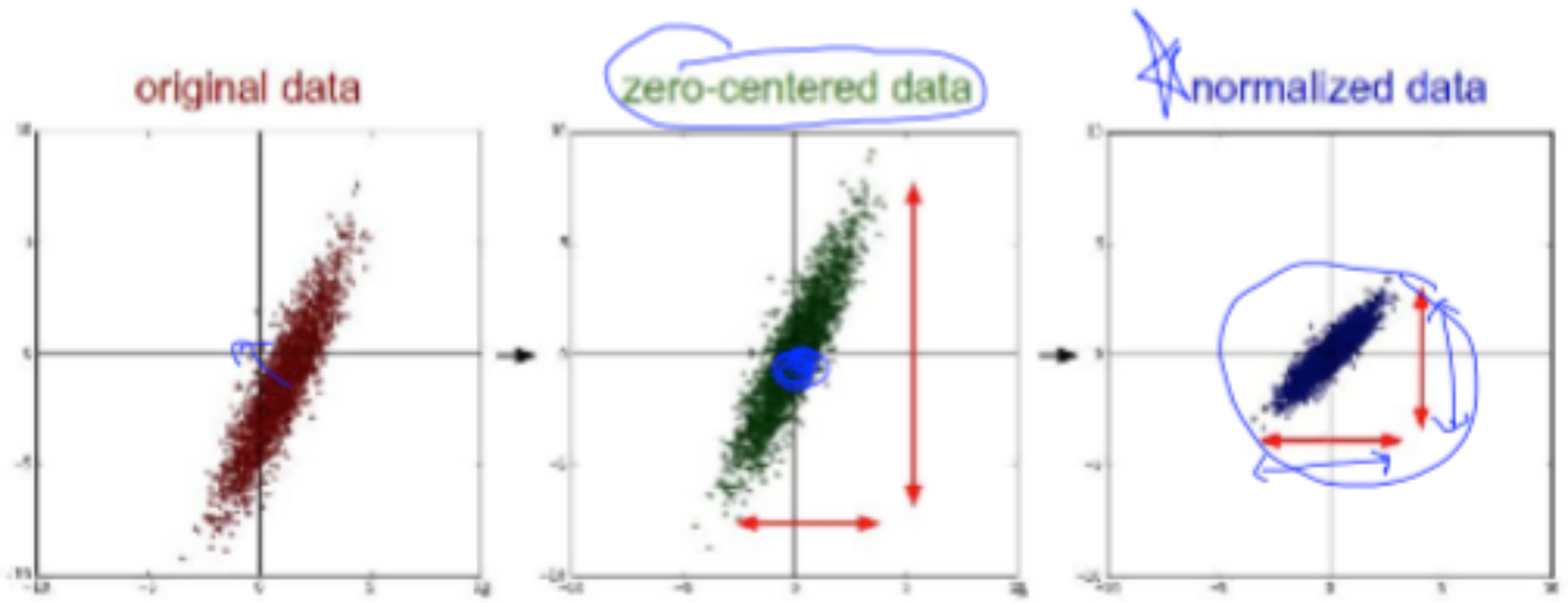
Data (X) preprocessing for gradient descent

x1	x2	y
1	9000	A
2	-5000	A
4	-2000	B
6	8000	B
9	9000	C



DATA PREPROCESSING FOR GRADIENT DESCENT

Data (X) preprocessing for gradient descent



STANDARDIZATION

Standardization

$$x'_j = \frac{x_j - \mu_j}{\sigma_j}$$

```
X_std[:,0] = (X[:,0] - X[:,0].mean()) / X[:,0].std()
```

http://sebastianraschka.com/Articles/2015_singlelayer_neurons.html

NORMALIZATION AND STANDARDIZATION

Normalization은 "해당 속성(변수)값-최소값/최대값-최소값"으로 0~1사이의 값으로 나타내는 척도법이고

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Standardization은 특정한 분포(ex. 정규분포)들의 평균과 분산 혹은 표준편차를 이용해 "속성값-평균/표준편차"로 해당 분포에서의 이 속성값이 평균으로부터의 위치를 표준편차 단위로 옮겨서 다시 나타낸 것이다.

$$x_{new} = \frac{x - \mu}{\sigma}$$

http://gentlej90.tistory.com/26

NORMALIZATION AND STANDARDIZATION

- **Normalization**

수식 : (요소값 - 최소값) / (최대값 - 최소값)

설명 : 전체 구간을 0~100으로 설정하여 데이터를 관찰하는 방법으로,

특정 데이터의 위치를 확인할 수 있게 해줌

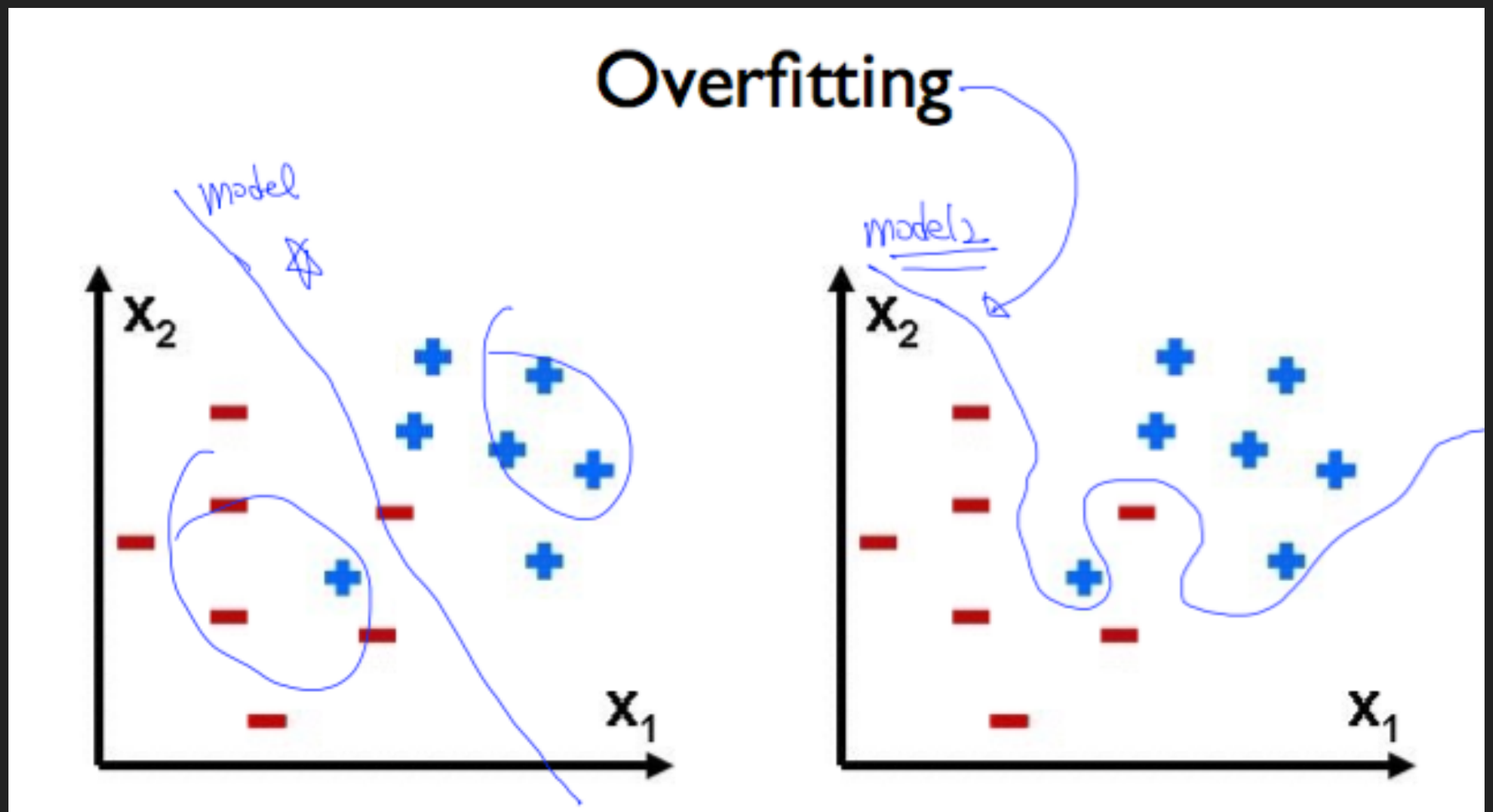
- **Standardization**

수식 : (요소값 - 평균) / 표준편차

설명 : 평균까지의 거리로, 2개 이상의 대상이 단위가 다를 때,

대상 데이터를 같은 기준으로 볼 수 있게 해줌

OVERFITTING



- Our model is very good with training data set (with memorization)
- Not good at test dataset or in real use

SOLUTIONS FOR OVERFITTING

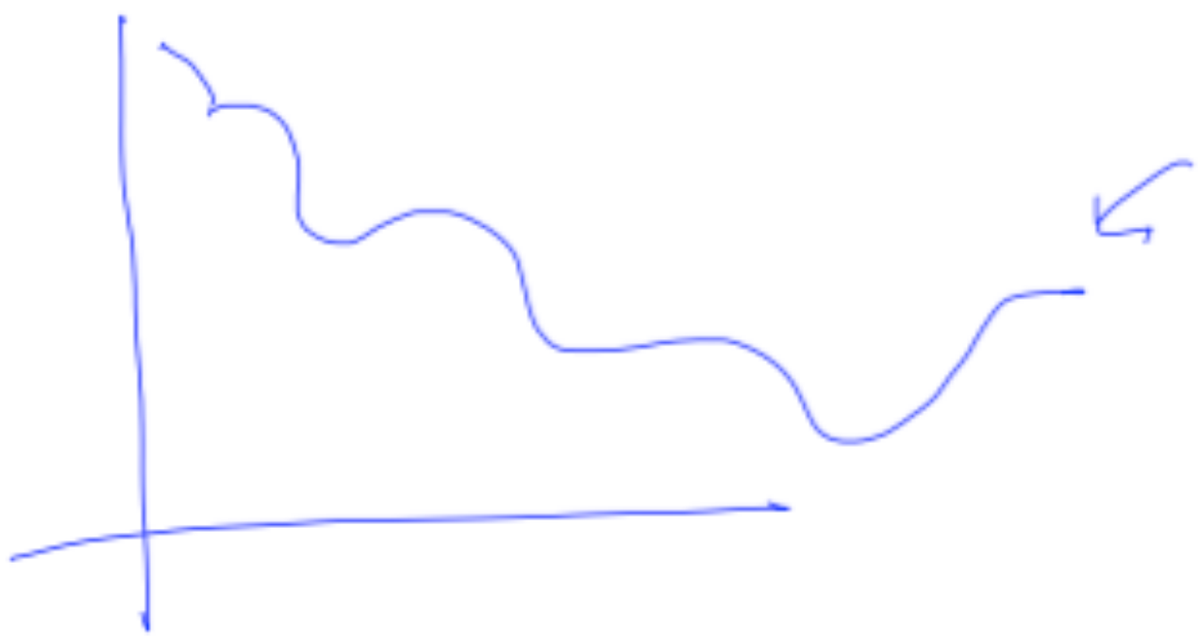
Solutions for overfitting

- More training data!
- Reduce the number of features
- Regularization

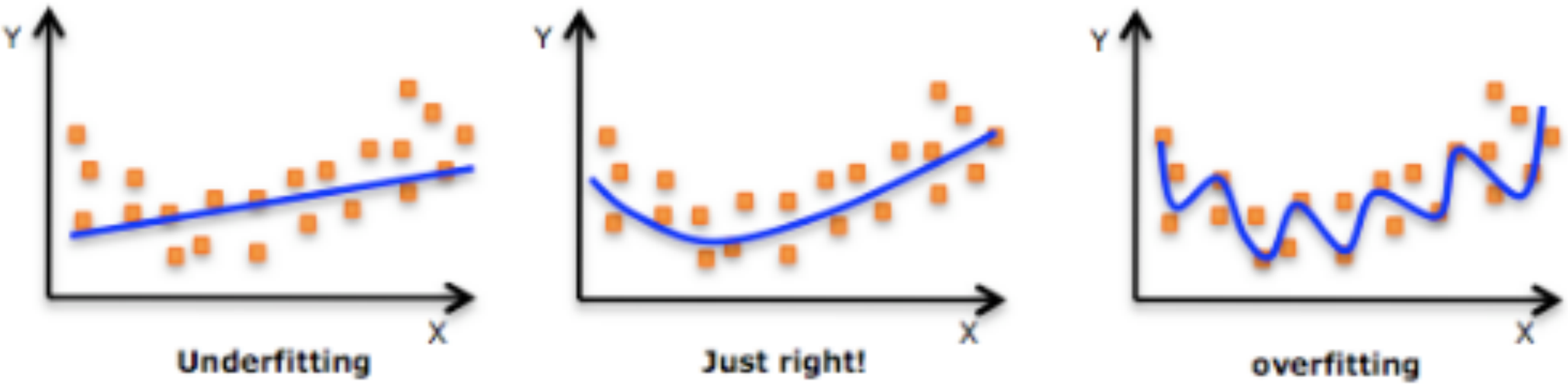
REGULARIZATION

Regularization

- Let's not have too big numbers in the weight



UNDERFITTING VS. OVERFITTING



REGULARIZATION FORMULA

Regularization

- Let's not have too big numbers in the weight

```
l2reg = 0.001 * tf.reduce_sum(tf.square(W))
```

$$L = \frac{1}{N} \sum_i D(S(Wx_i + b), L_i) + \lambda \sum W^2$$

LOSS

TRAINING SET

PERFORMANCE EVALUATION : IT THIS GOOD?

Performance evaluation: is this good?

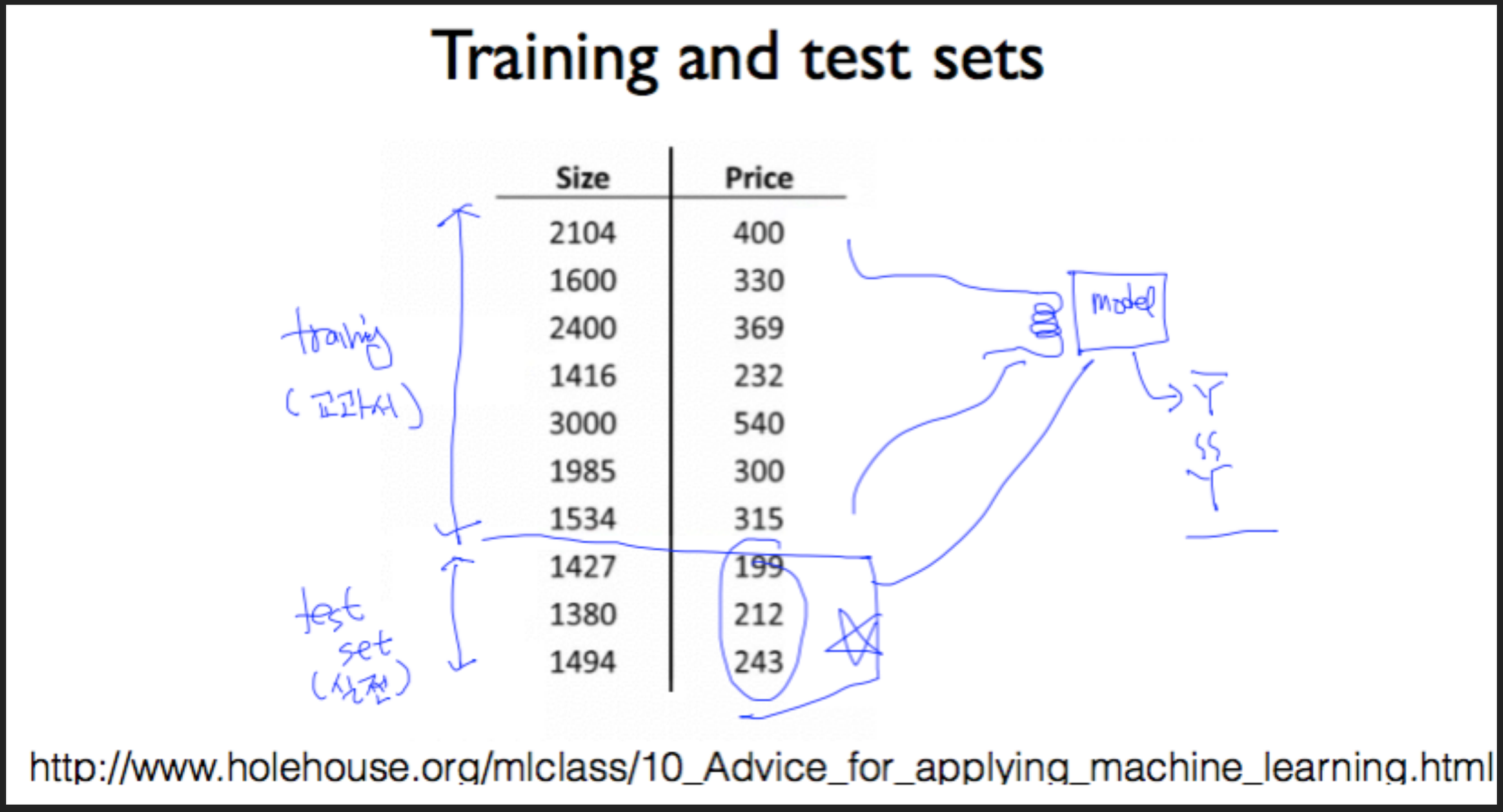
EVALUATION USING TRAINING SET

Evaluation using training set?

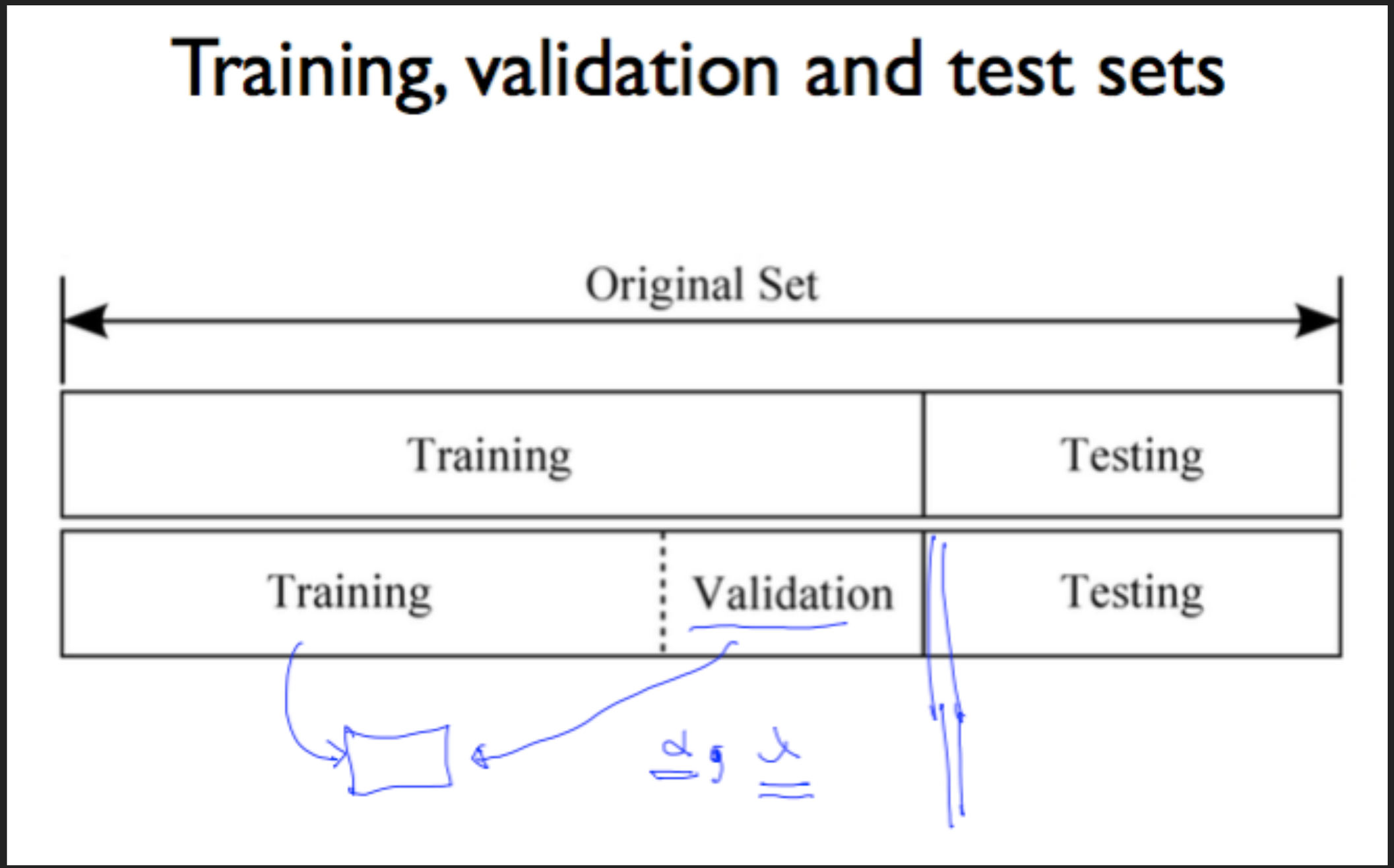
Size	Price
2104	400
1600	330
2400	369
1416	232
3000	540
1985	300
1534	315
1427	199
1380	212
1494	243

- 100% correct (accuracy)
- Can memorize

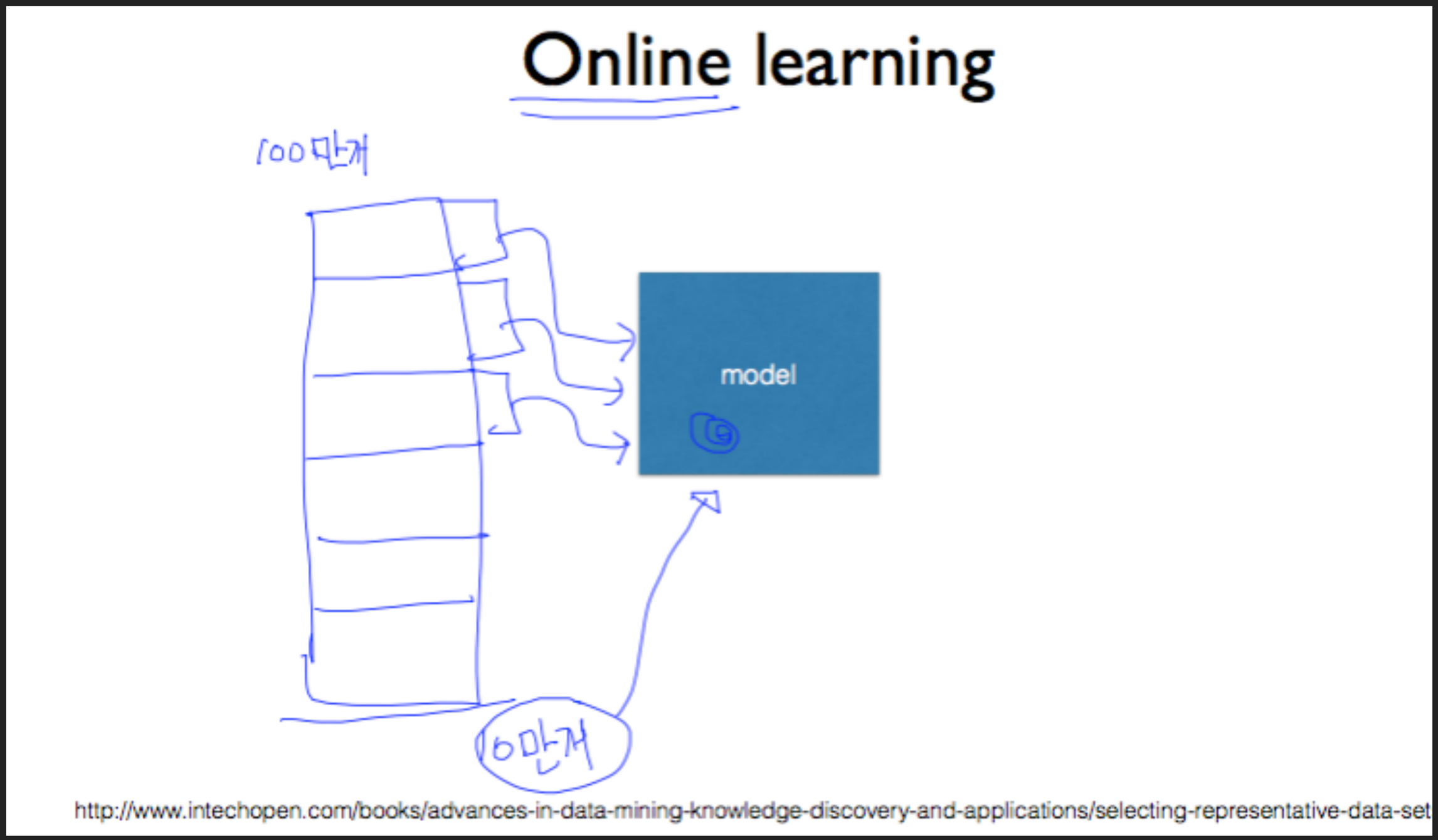
TRANINING AND TEST SETS



TRAINING, VALIDATION AND TEST SETS



ONLINE LEARNING



http://www.intechopen.com/books/advances-in-data-mining-knowledge-discovery-and-applications/selecting-representative-data-set

MNIST DATASET