Before & After

The AI Winter

Session #1: Before the AI Winter

Microsoft Student Partners

Evangelist Younjoon Chung

Goals

- Introduction to machine learning

- Basic understanding of machine learning algorithms
  - Linear regression, Logistic regression
  - Gradient Descent Algorithm
  - Sigmoid function
  - Softmax Classifier
  - Overfitting, Regularization

Acknowledgement

- Andrew Ng's ML class
  - https://class.coursera.org/ml-003/lecture
  - http://www.holehouse.org/mlclass (note)
- 모두의 딥러닝
  - https://hunkim.github.io/ml
- The Deep Learning textbook
  - http://www.deeplearningbook.org
- 김지환 교수님 System programming
- http://www.andreykurenkov.com/writing/a-brief-history-of-neural-nets-and-deep-learning

Machine Learning

- Limitations of explicit programming
  - Spam filter: many rules
  - Automatic driving: too many rules

- Machine learning: "Field of study that gives computers the ability to learn without being explicitly programmed" Arthur Samuel (1959)

What does it mean for machines to learn?

- 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***. Tom M. Mitchell (1997)

- ***E***: experience
- ***T***: task
- ***P***: performance measure

The Task, ***T***

- The process of learning itself is not the task.

- Usually described in terms of **how the machine learning system should process a [collection of features].** (“Deep learning” 99)

What are the tasks?

- **Classification**
  - Learning a function $\boldsymbol{f}: \boldsymbol{R^n} \rightarrow \{\boldsymbol{1} \ \ldots \boldsymbol{k}\}$
  - When $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$, the model assigns an input described by vector $\boldsymbol{x}$ to a category identified by numeric code $\boldsymbol{y}$ ("Deep learning" 100)
  - Outputs can also be probability distribution over classes.

- Examples
  - Object recognition
  - Spam detector
  - Face recognition -> Viola Jones!
  - MNIST

What are the tasks?

- **Regression**
  - Learning a function $\boldsymbol{f}: \boldsymbol{R^n} \rightarrow \boldsymbol{R}$
  - Outputs a numerical value (continuous)
  - Similar to classification; only the format of the output is different

- Examples
  - Weather forecast
  - Predict a test score (if the score is mapped to grades, then it is a classification!)

What are the tasks?

- **Translation**
  - Different from transcription in that the input values are already in a structured format.
  - Machines are asked to convert a sequence of symbols to another language.

- Examples
  - Google Translation

What are the tasks?

- **Anomaly detection**
  - Sifts through a set of events or objects, and flags some of them as being unusual or atypical. (“Deep Learning” 102)

- Examples
  - Credit card fraud detection. By modeling your purchasing habits, a credit card company detect misuse of your cards.

The Performance Measure, ***P***

- The choice of P depends on the nature of the problems.
  - We have to first decide what should be measured.
    - Should we penalize the system more if it frequently makes medium-sized mistakes and rarely makes very large mistakes, and vice versa.
  - how they should be measured
    - 0-1 loss, cross-entropy, MSE, etc.

- Usually we are interested in how well the machine learning algorithm performs on data it has not seen before (“Deep Learning” 104)

The Experience, ***E***

- Machine learning algorithms can be broadly categorized as **unsupervised** or **supervised** by what kind of experience they are allowed to have during the learning process. (“Deep Learning” 104)

- Supervised learning algorithms
  - Dataset with labels -> temperature prediction

- Unsupervised learning algorithms
  - Dataset without labels -> hash tag clustering

Supervised learning

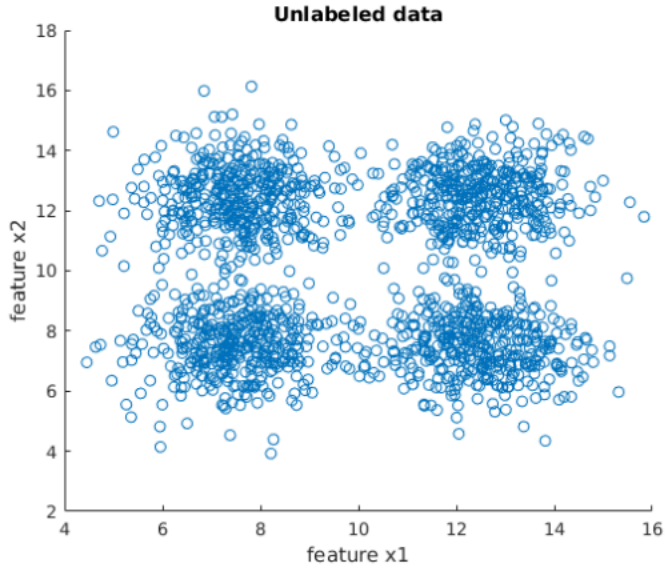
- Supervised learning
  - learning with labeled examples - training set

- An example training set for four visual categories.

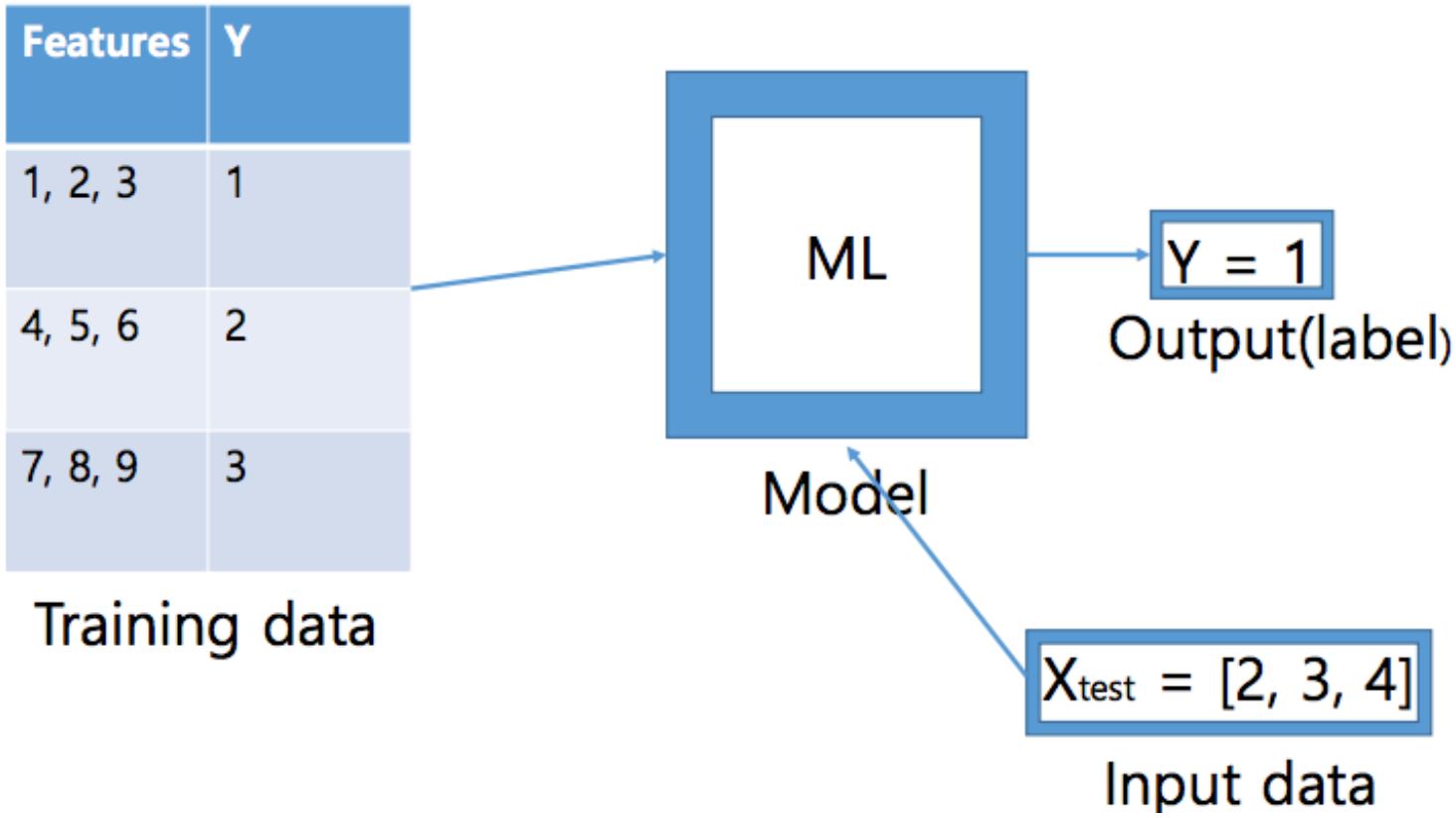
Unsupervised learning

- Unsupervised learning: un-labeled data
  - Google news grouping
  - Word clustering

- An example of clustering



Supervised learning in a nutshell



Types of supervised learning

- Predicting final exam score based on time spent
  - **Regression**

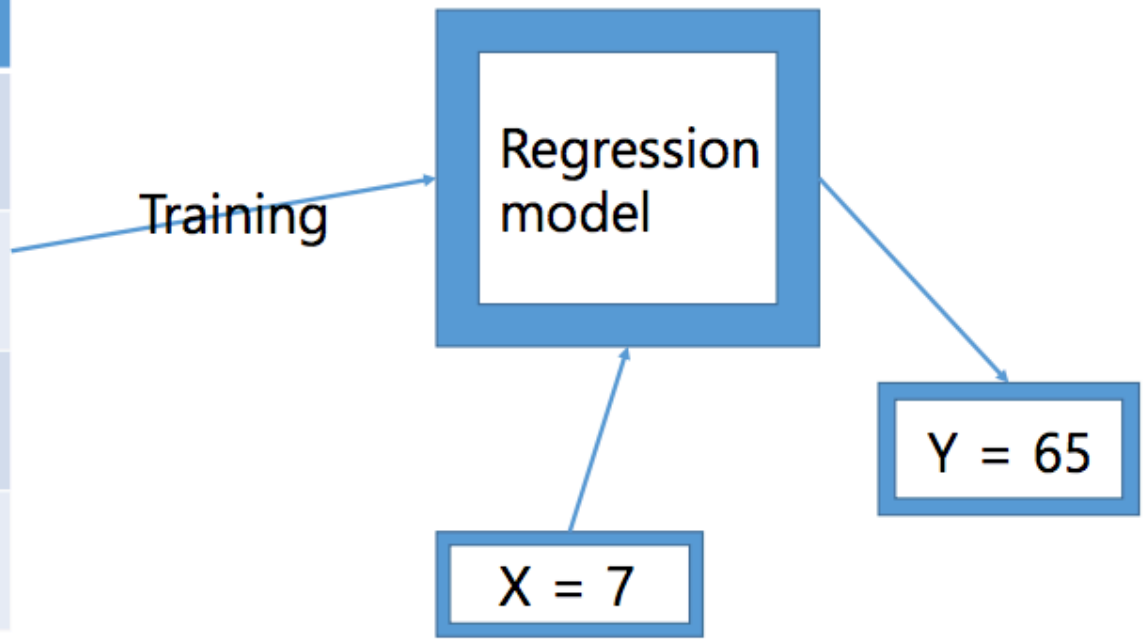
- Pass/non-pass based on time spent
  - **Binary classification**

- Letter grade(A, B, C, D, and F) based on time spent
  - **Multi-label classification**

Predicting exam score: regression

0~100

X(hours)	y(score)
10	90
9	80
3	50
2	30

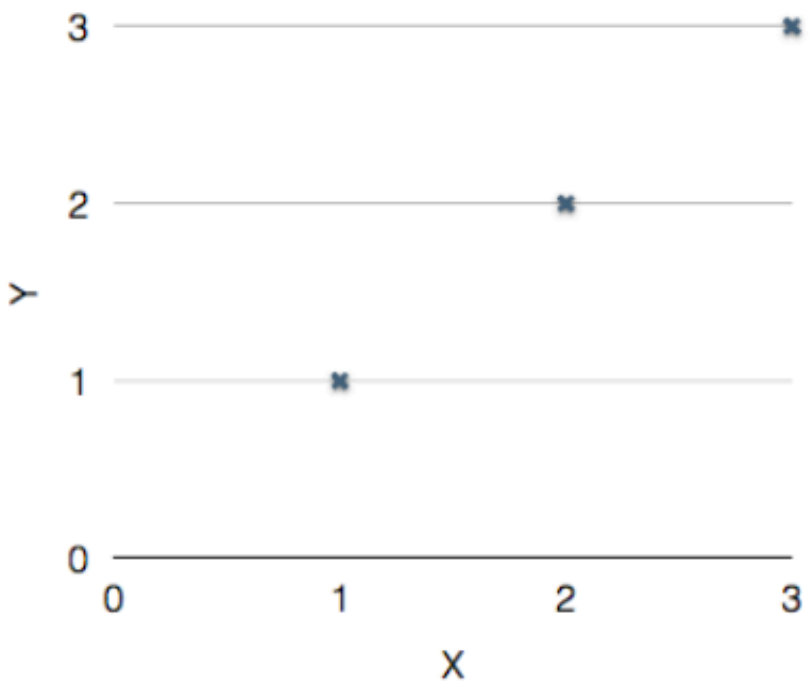


Regression (data)

X	Y
1	1
2	2
3	3

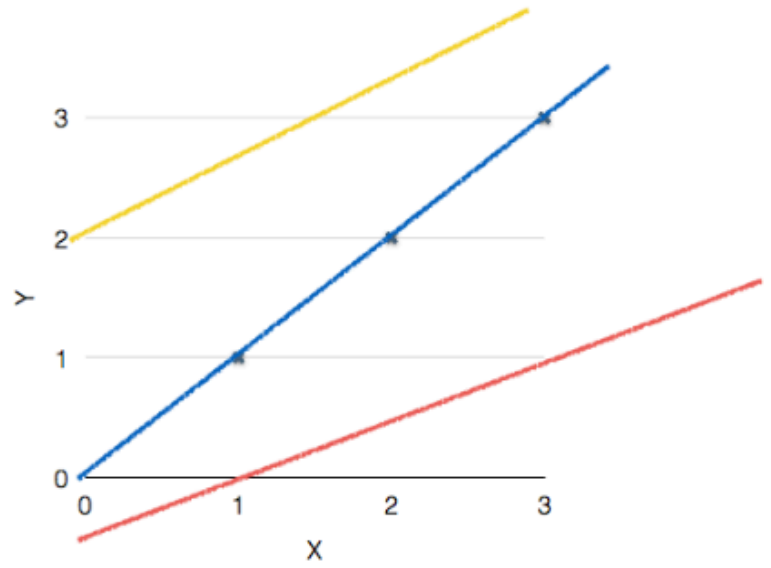
Regression (presentation)

X	Y
1	1
2	2
3	3



(Linear) Hypothesis

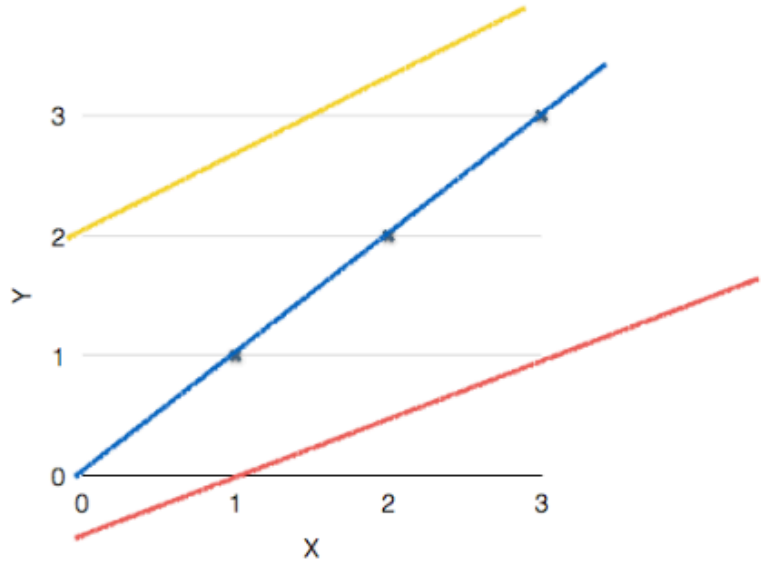
$$H(x) = Wx + b$$



(Linear) Hypothesis

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

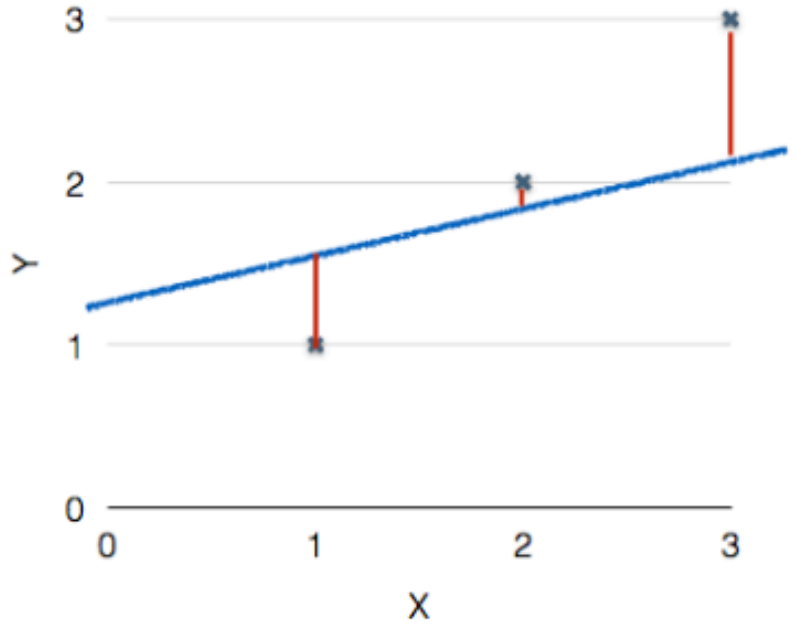
Which hypothesis is better?



Which hypothesis is better?

- We need a measure for deciding which hypothesis is most accurate!

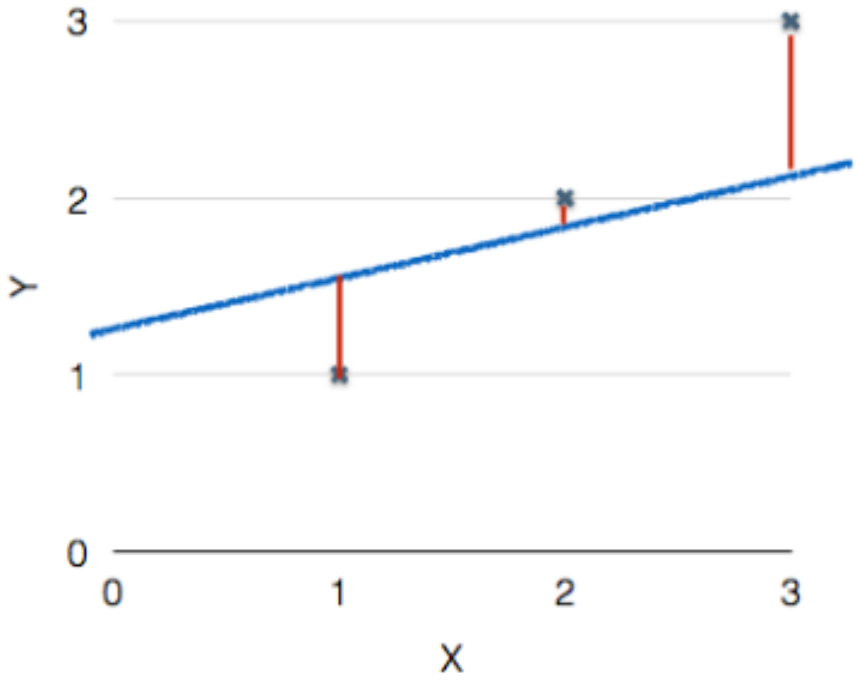
-> use differences between prediction & real value



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 function in a nutshell

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

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

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

Cost function is now a function of
W, b(weight, bias)!

- Training Goal:

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

Lower the cost,
better the
hypothesis(model)!

Simplified hypothesis

Eliminated bias for educational purposes

$$H(x) = Wx$$

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

Now the cost function is a function of only W

What does cost(W) look 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) = ?

What does cost(W) look like?

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

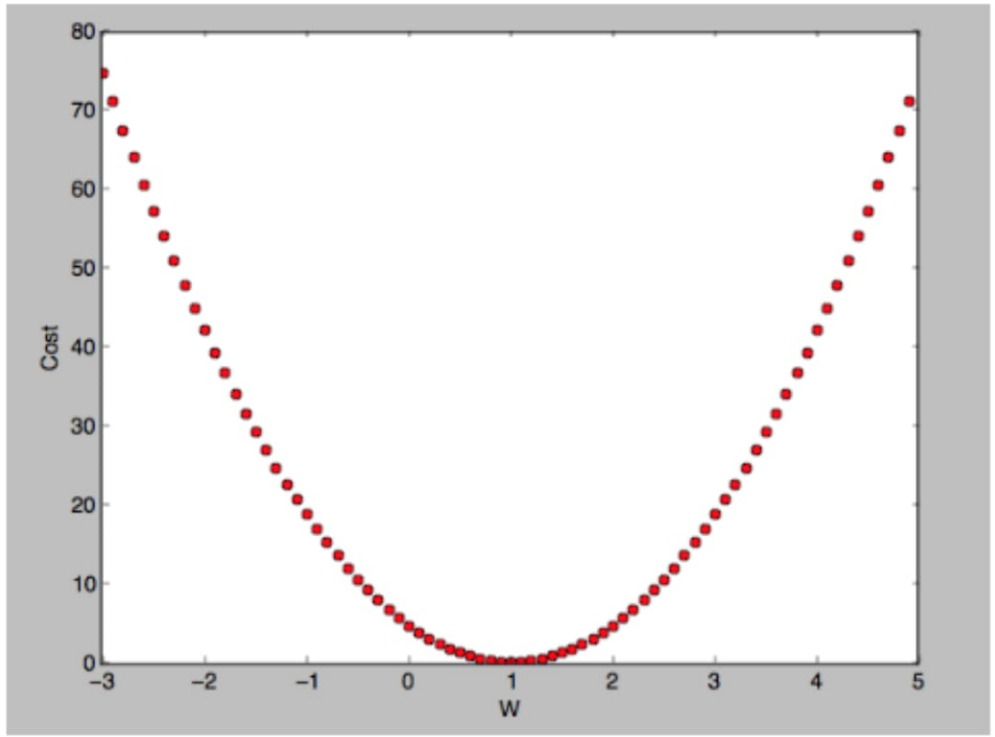
X	Y
1	1
2	2
3	3

- 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
- W = 2, cost(W) = 4.67

What does cost(W) look like?

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

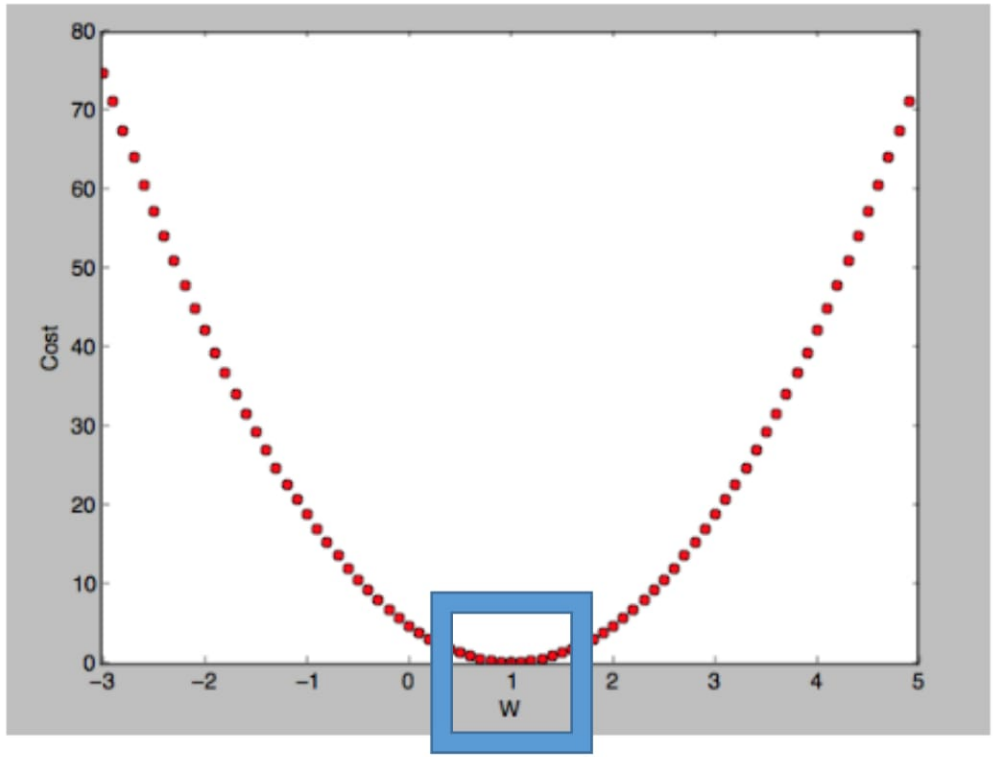


What does cost(W) look like?

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

How would you find the lowest point?

Finding root of derivative not computationally feasible with large number of features!



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 does gradient descent work?

- 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

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

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

- Updating weights

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

Learning rate, *determines the step size*

gradient

Formal definition

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

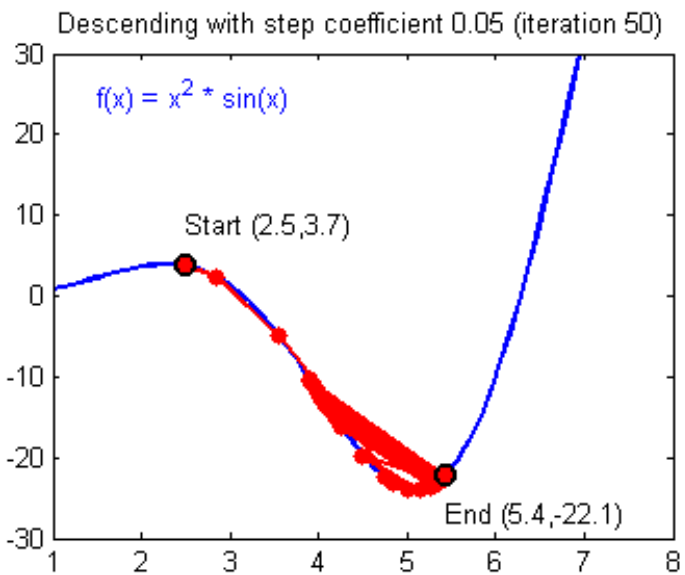
$$W := W - \alpha\frac{\partial}{\partial W}cost(W)$$

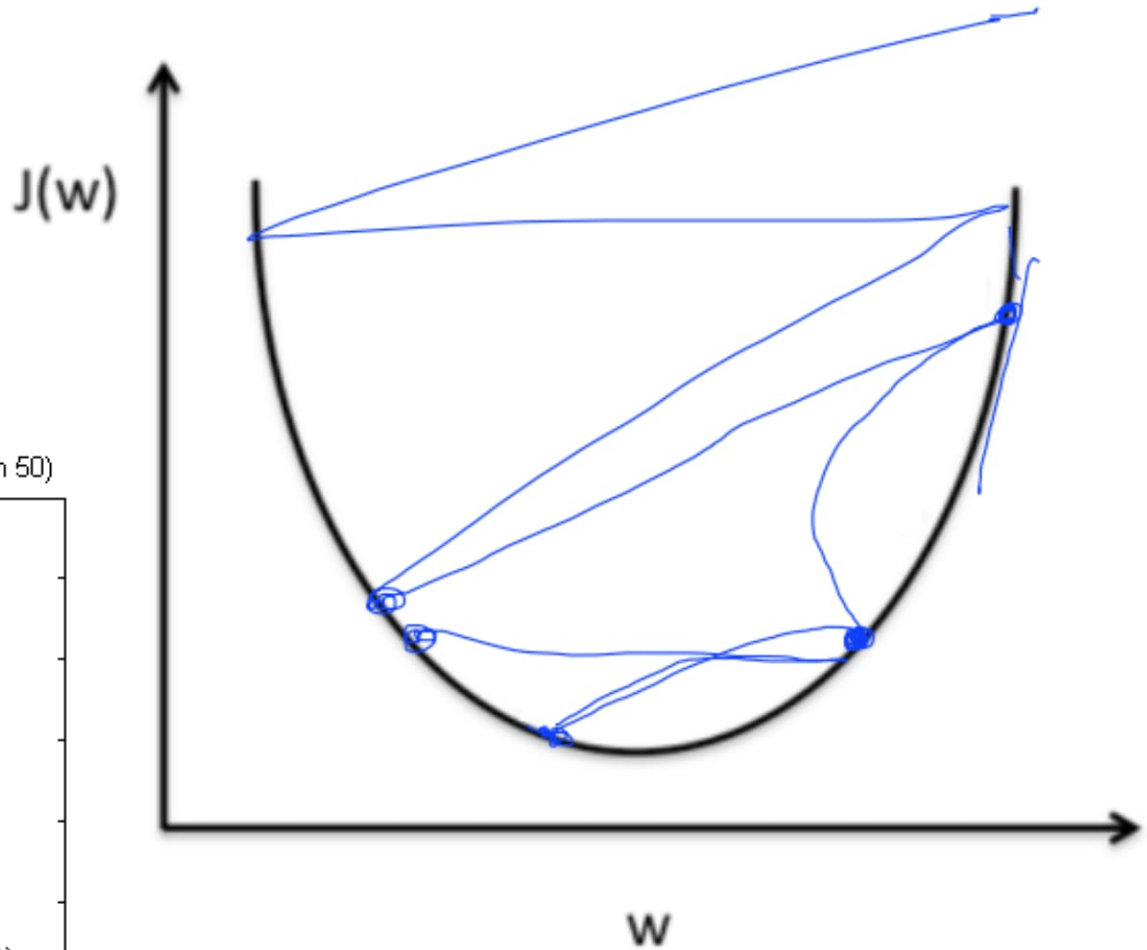
$$W := W - \alpha\frac{\partial}{\partial W}\frac{1}{2m}\sum_{i=1}^{m}(Wx^{(i)} - y^{(i)})^2$$

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

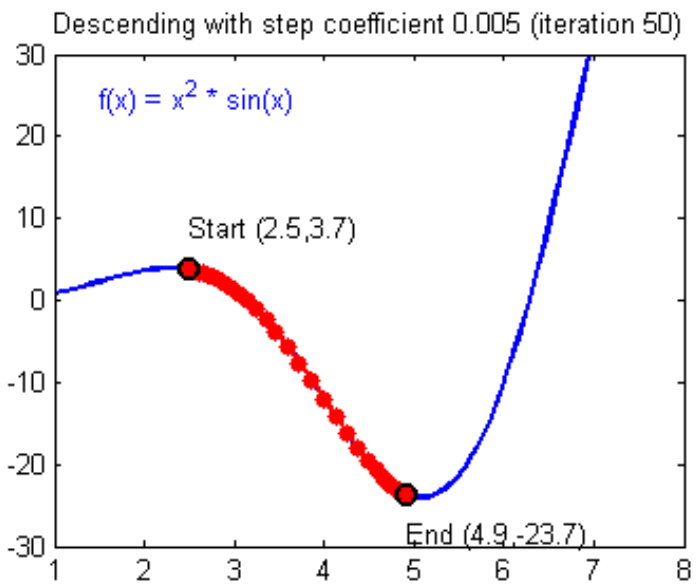
$$W := W - \alpha\frac{1}{m}\sum_{i=1}^{m}(Wx^{(i)} - y^{(i)})x^{(i)}$$

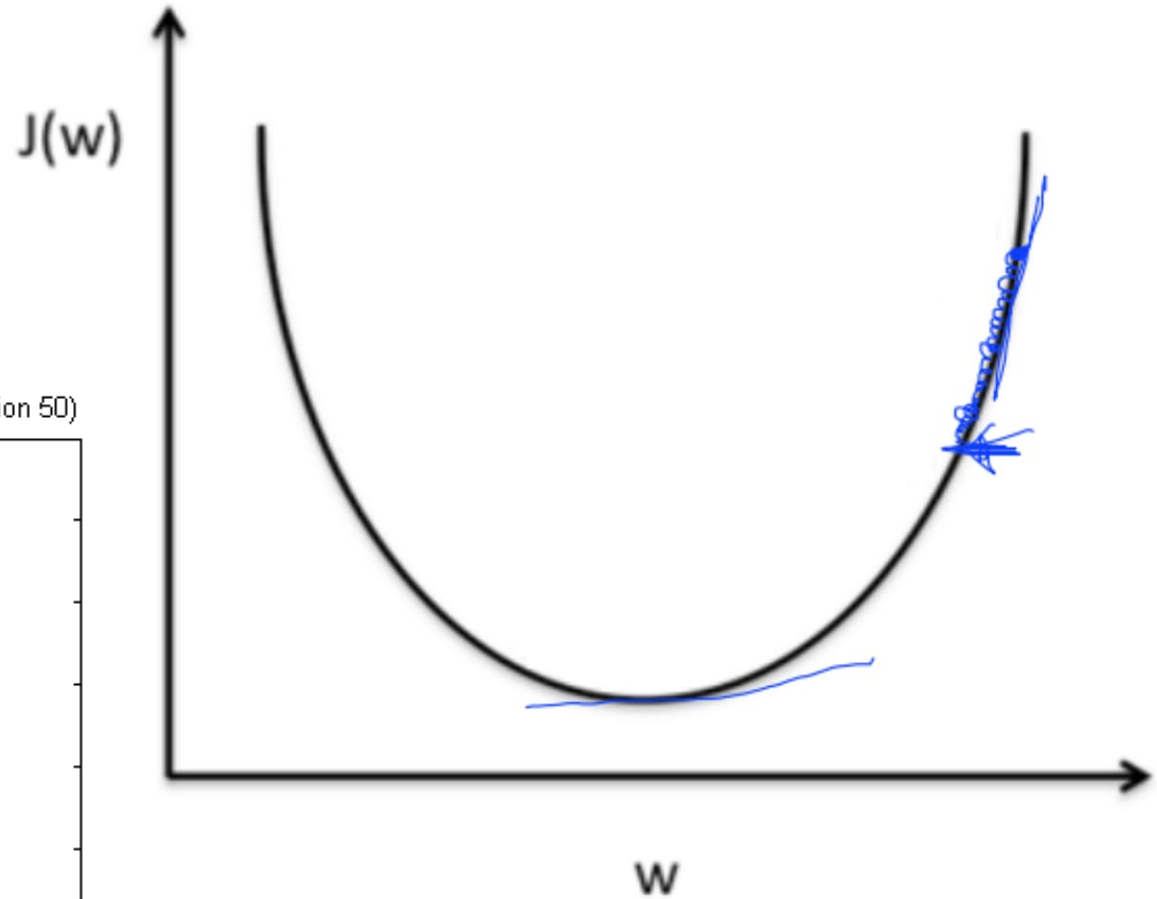
Issues:
large learning rate - overshooting





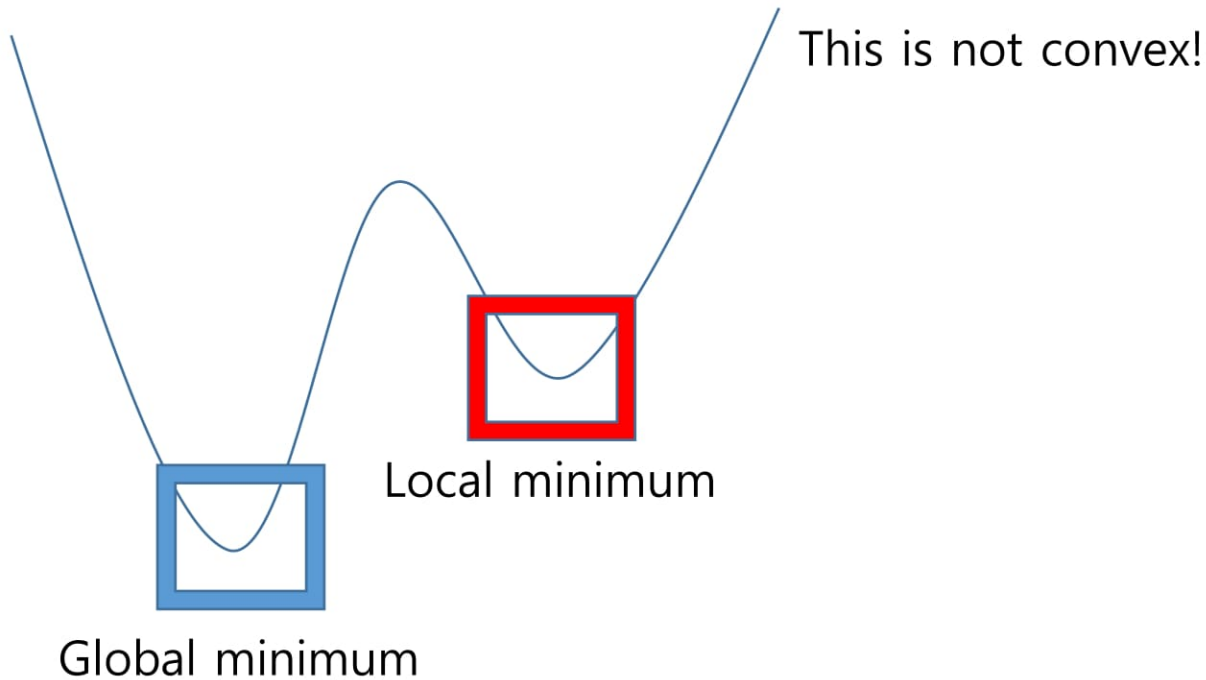
Issues:
small learning rate - takes too long





Issues:
starting point is important

- Only works on convex functions!
- Can end up in local minimum depending on the starting point



Issues:
only works on differentiable functions

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

We take the derivative of *cost(W),*
therefore cost(W) must be differentiable

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

By chain rule,
the hypothesis must also be differentiable

$$H(x) = Wx$$

Predicting exam score: regression using three inputs (x1, x2, x3)

multi-variable/feature

x_1 (quiz 1)	x_2 (quiz 2)	x_3 (midterm 1)	Y (final)
73	80	75	152
93	88	93	185
89	91	90	180
96	98	100	196
73	66	70	142

Test Scores for General Psychology

For multiple features

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

this is for a single feature

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

$$H(x_1, x_2, x_3) = w_1x_1 + w_2x_2 + w_3x_3 + b$$

$$cost(W, b) = \frac{1}{m}\sum_{I=1}^{m}(H(x_1^{(i)}, x_2^{(i)}, x_3^{(i)}) - y^{(i)})^2$$

Recap

- (Linear) Hypothesis

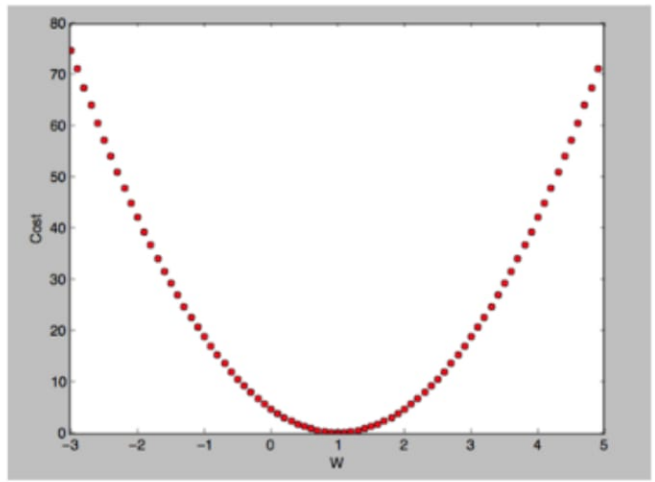
$$H(x_1, x_2, x_3, ..., x_n) = w_1 x_1 + w_2 x_2 + w_3 x_3 + ... + w_n x_n + b$$

- Cost function

$$cost(W, b) = \frac{1}{m} \sum_{I=1}^{m} (H(x_1^{(i)}, x_2^{(i)}, x_3^{(i)}) - y^{(i)})^2$$

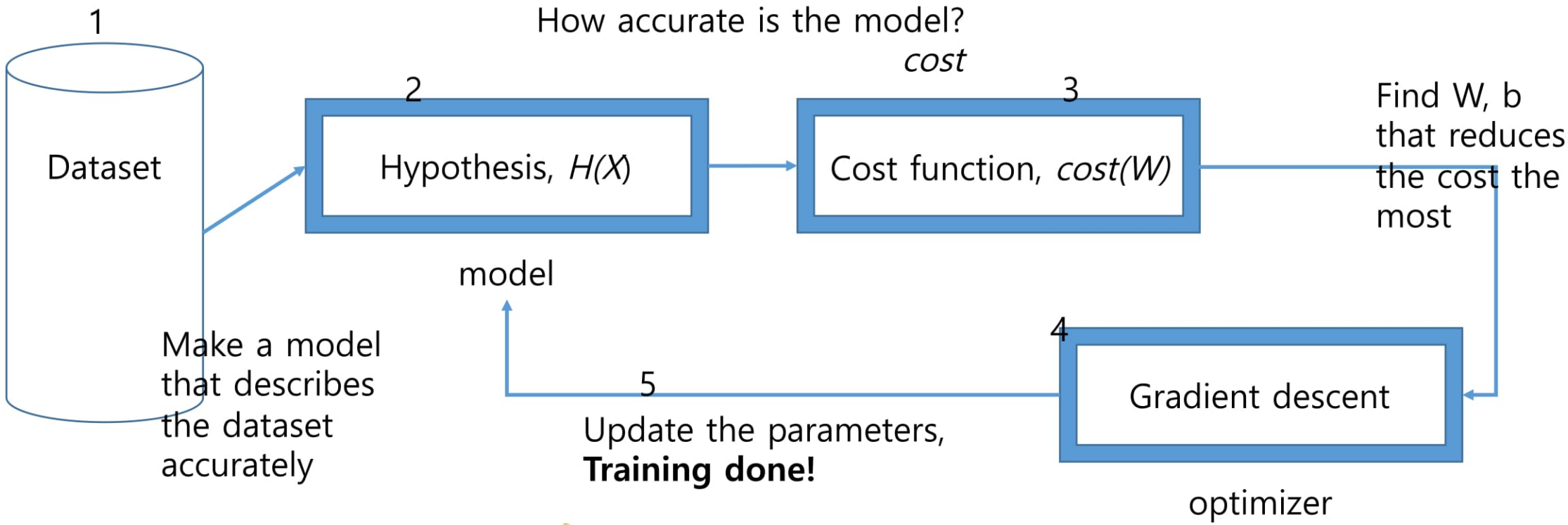
- Gradient descent algorithm

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



Cost function for single variable hypothesis

Training in a nutshell



So module! Many simple!
Wow.

(Review) Types of supervised learning

- Predicting final exam score based on time spent
  - **Regression**
- Pass/non-pass based on time spent
  - **Binary classification**
- Letter grade(A, B, C, D, and F) based on time spent
  - **Multi-label classification**

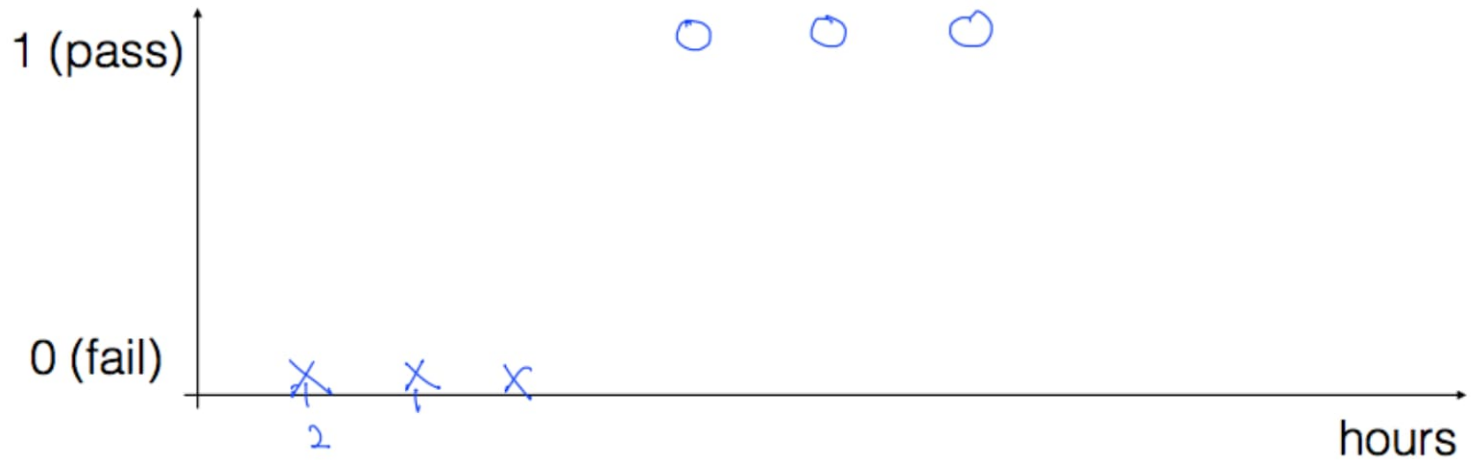
(Binary) Classification

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

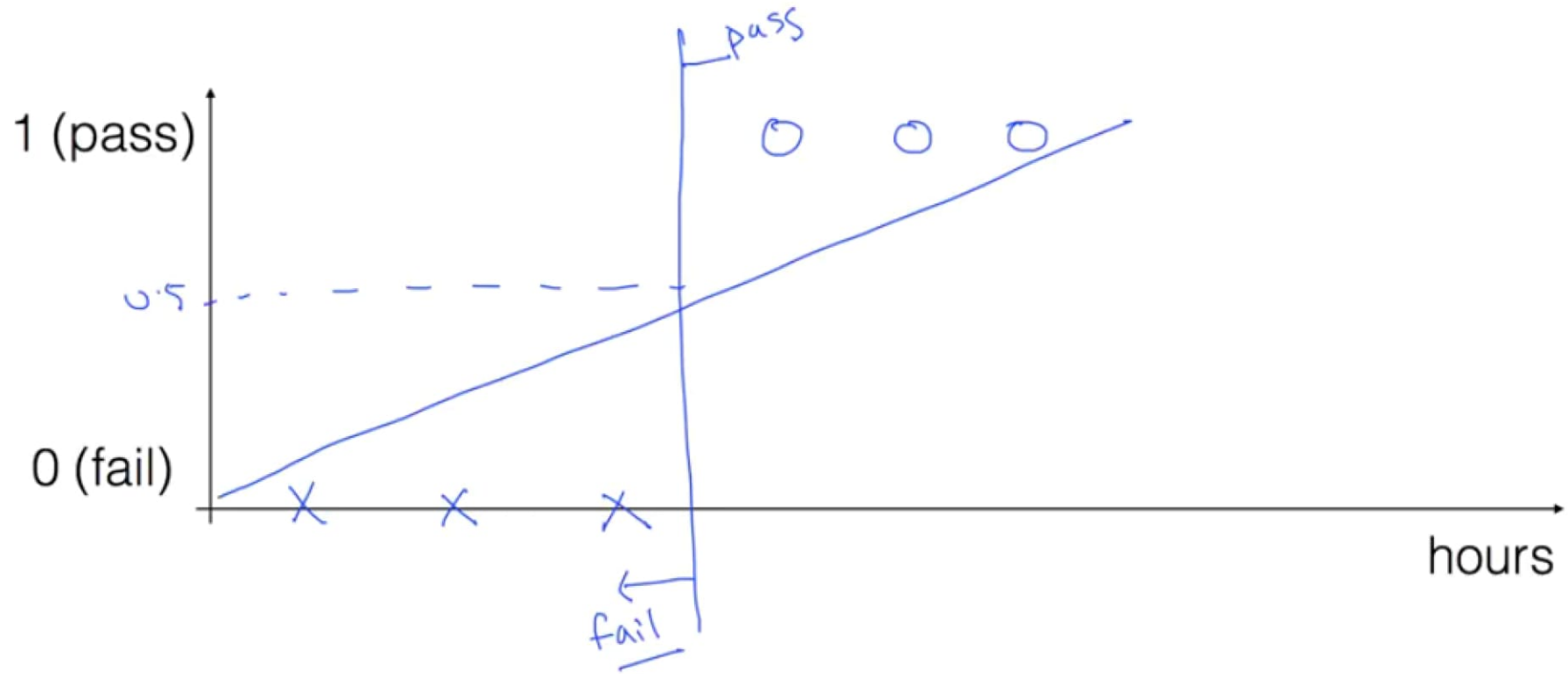
0, 1 encoding

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

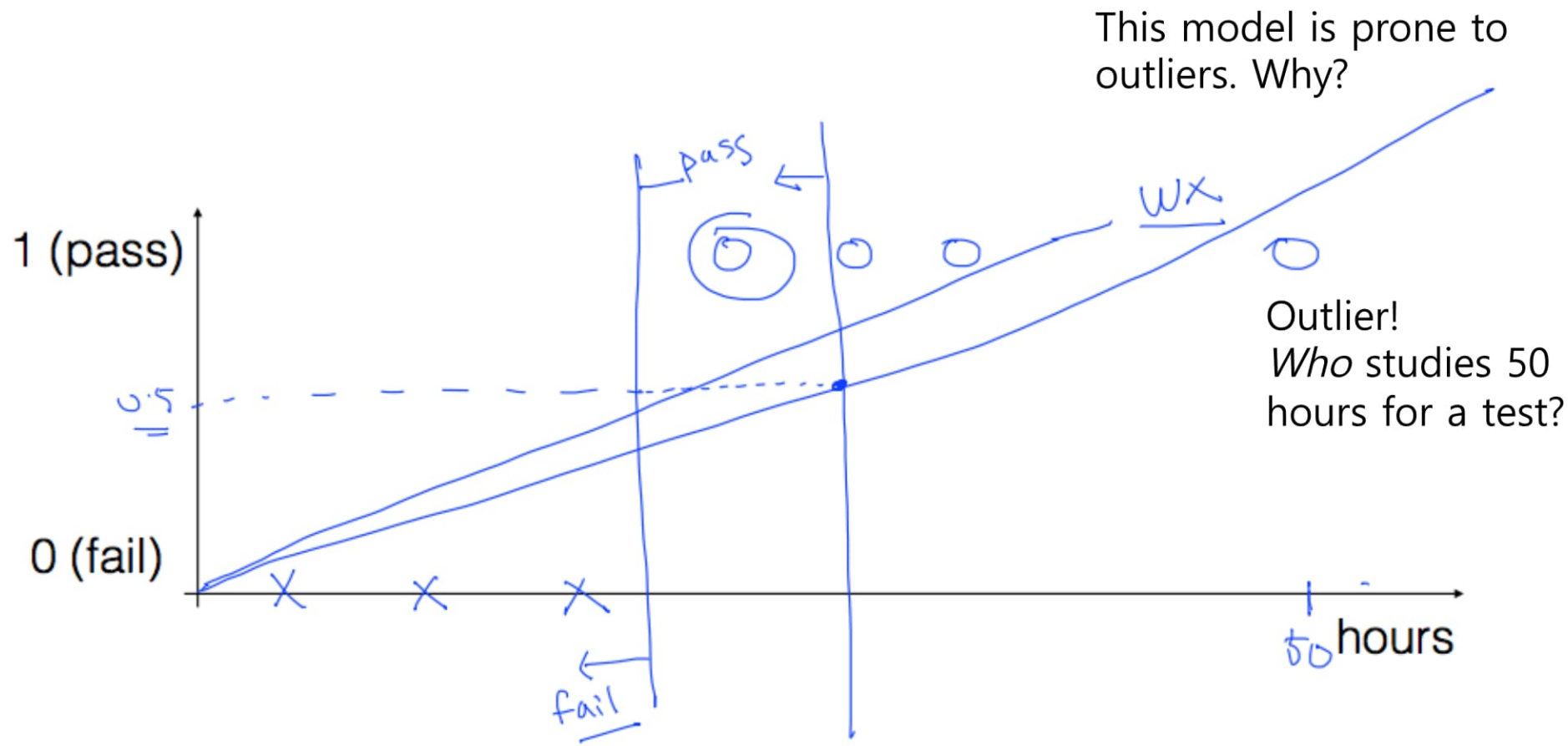
Pass(1)/Fail(0) based on study hours



Linear regression?



Linear regression?



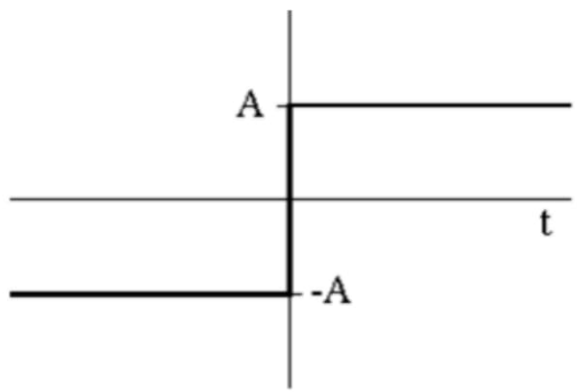
We need conversion!

- We know Y is 0 or 1, and H(X) outputs values larger than 1 or less than 0

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

We want to limit the output range to be within 0 ~ 1

How about this?



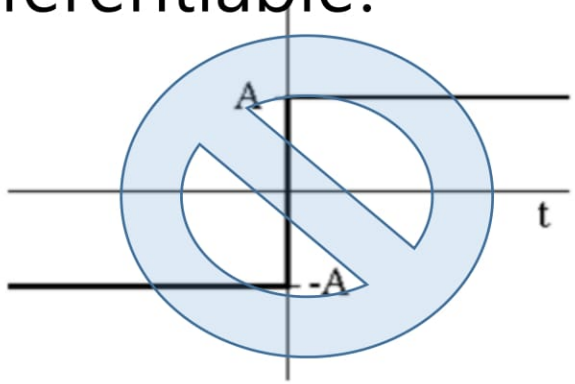
We need conversion!

- We know Y is 0 or 1, and H(X) outputs values larger than 1 or less than 0

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

We want to limit the output range to be within 0 ~ 1

- It has to be differentiable!

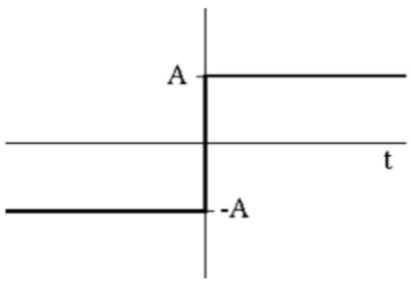


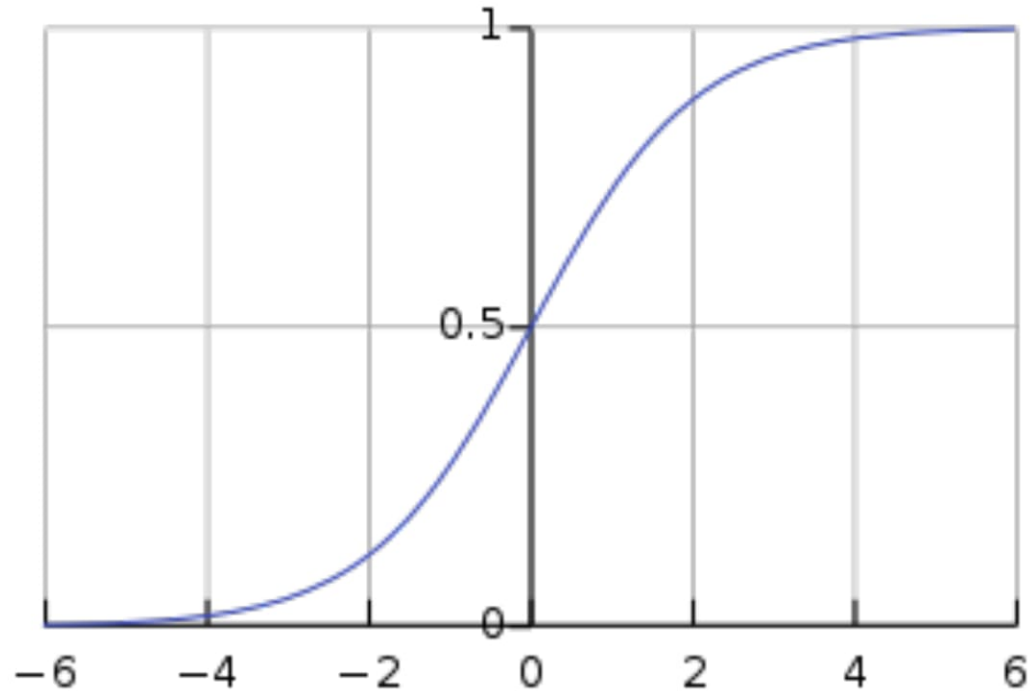
Plus, it's derivative is 0 no matter what x.
This poses a big problem later on.

Sigmoid

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

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





We need conversion!

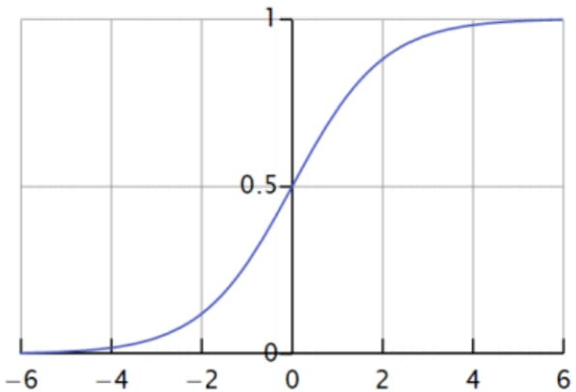
- We know Y is 0 or 1, and H(X) outputs values larger than 1 or less than 0

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

We want to limit the output range to be within 0 ~ 1

- It has to be differentiable!

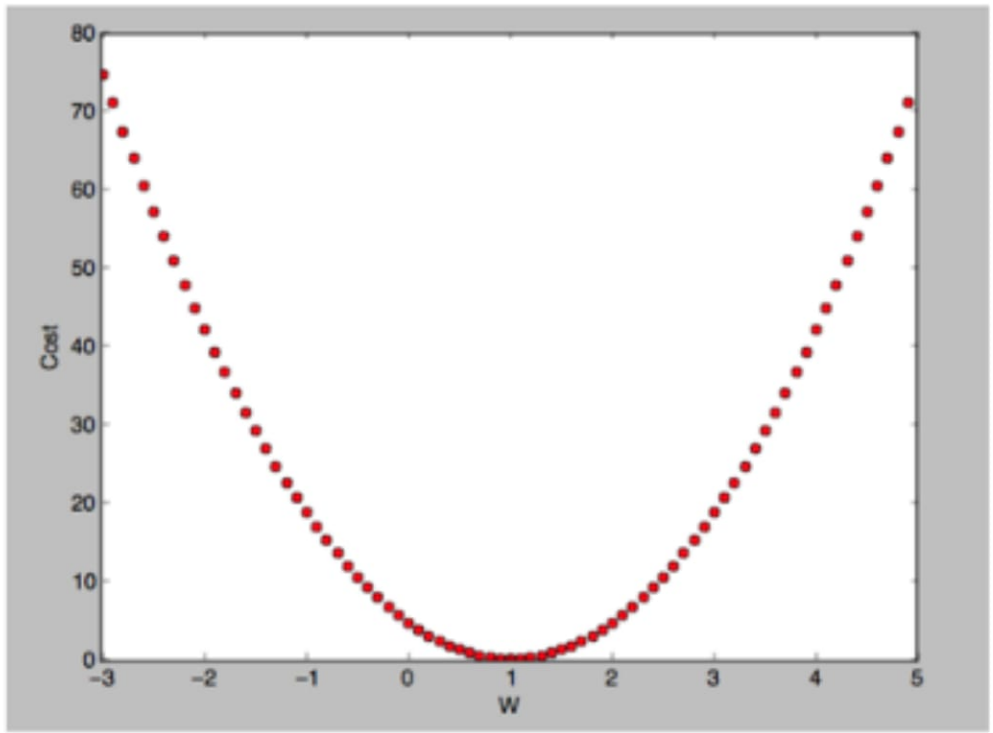
$$H(X) = \frac{1}{1 + e^{-W^T X}}$$



Sigmoid!

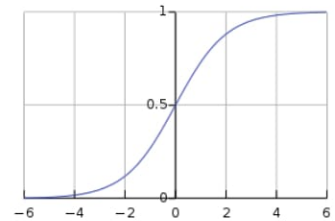
(Review) Cost

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



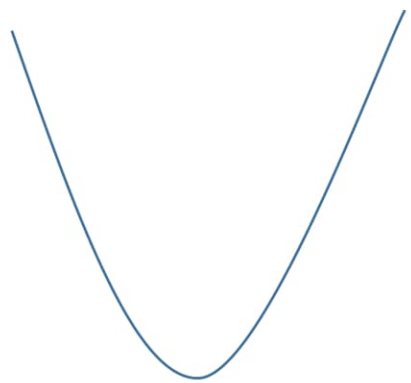
Cost function

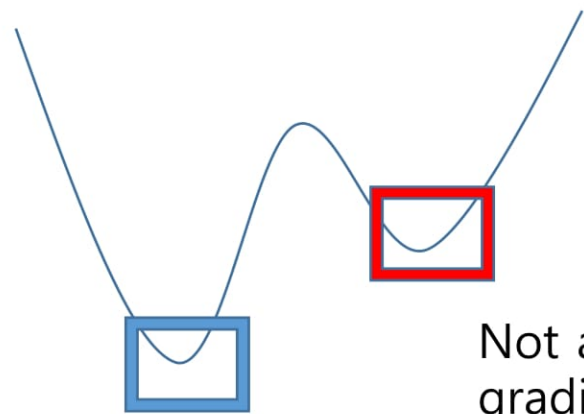
$$cost(W,b) = \frac{1}{m}\sum_{i=1}^{m}(H(x^{(i)}) - y^{(i)})^2$$



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

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





Not appropriate for gradient descent algorithm!

New cost function for logistic

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

Real value(answer, label)

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

Understanding cost function

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

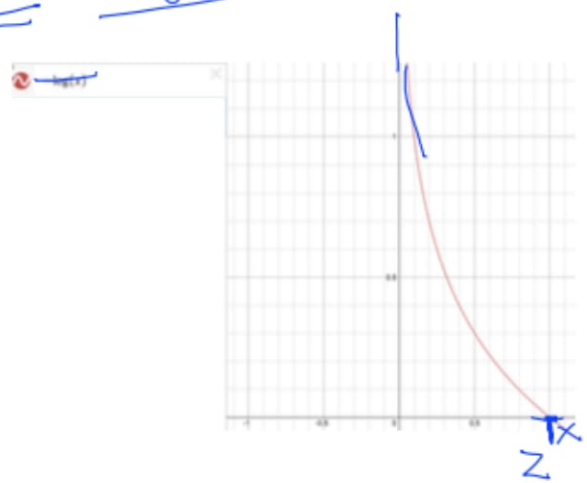
$\frac{1}{1+e^{-z}}$ log

Cost = $-\log(z)$

$y=1$

$H(x)=1 \rightarrow cost(1)=0$

$H(x)=0 \rightarrow cost=\infty$

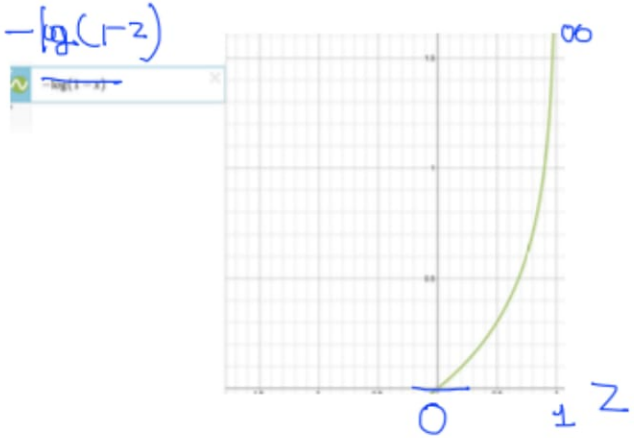


$y=0$

$H(x)=0$, $cost=0$

$H(x)=1$, $cost=\infty$

$-\log(1-z)$



New cost function for logistic

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

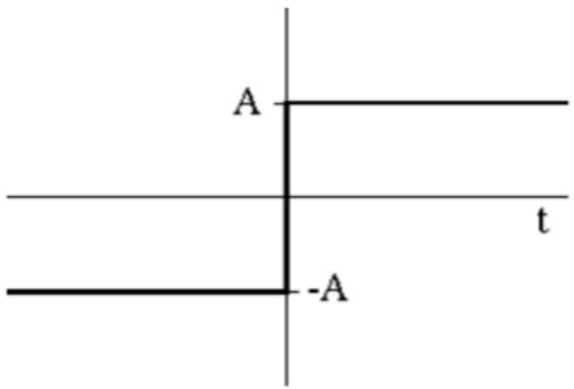
$$c(H(x), y) = \begin{cases} -log(H(x)) & : y = 1 \\ -log(1 - H(x)) & : y = 0 \end{cases}$$

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

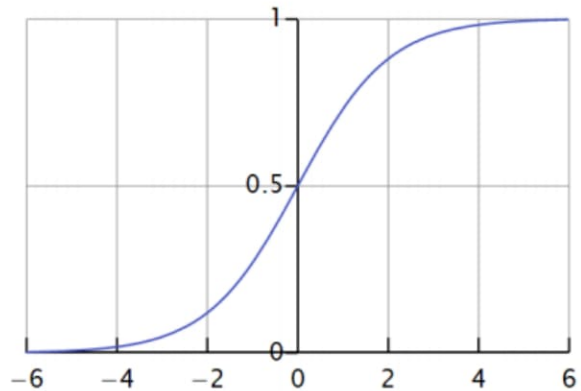
Minimize cost with gradient descent algorithm

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

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

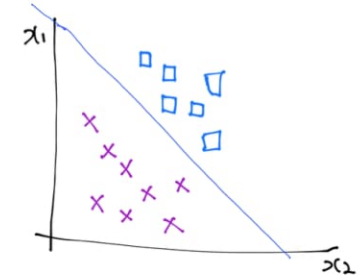


vs



Why sigmoid?

Training in a nutshell



1

Dataset

2

H(X)

Linear model

3

Sigmoid

Activation function

How accurate is the model?

cost

4

Cost function, *cost(W)*

Find W, b that reduces the cost the most

5

Gradient descent

optimizer

6

Update the parameters,
Training done!

Make a model that describes the dataset accurately

So module! Many simple!
Wow.

WOW

(Review) Types of supervised learning

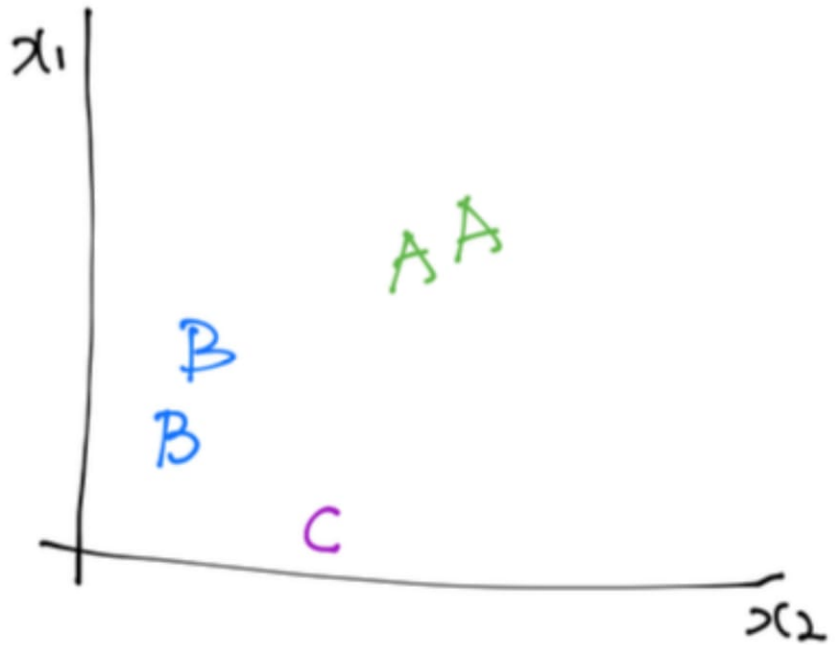
- Predicting final exam score based on time spent
  - **Regression**

- Pass/non-pass based on time spent
  - **Binary classification**

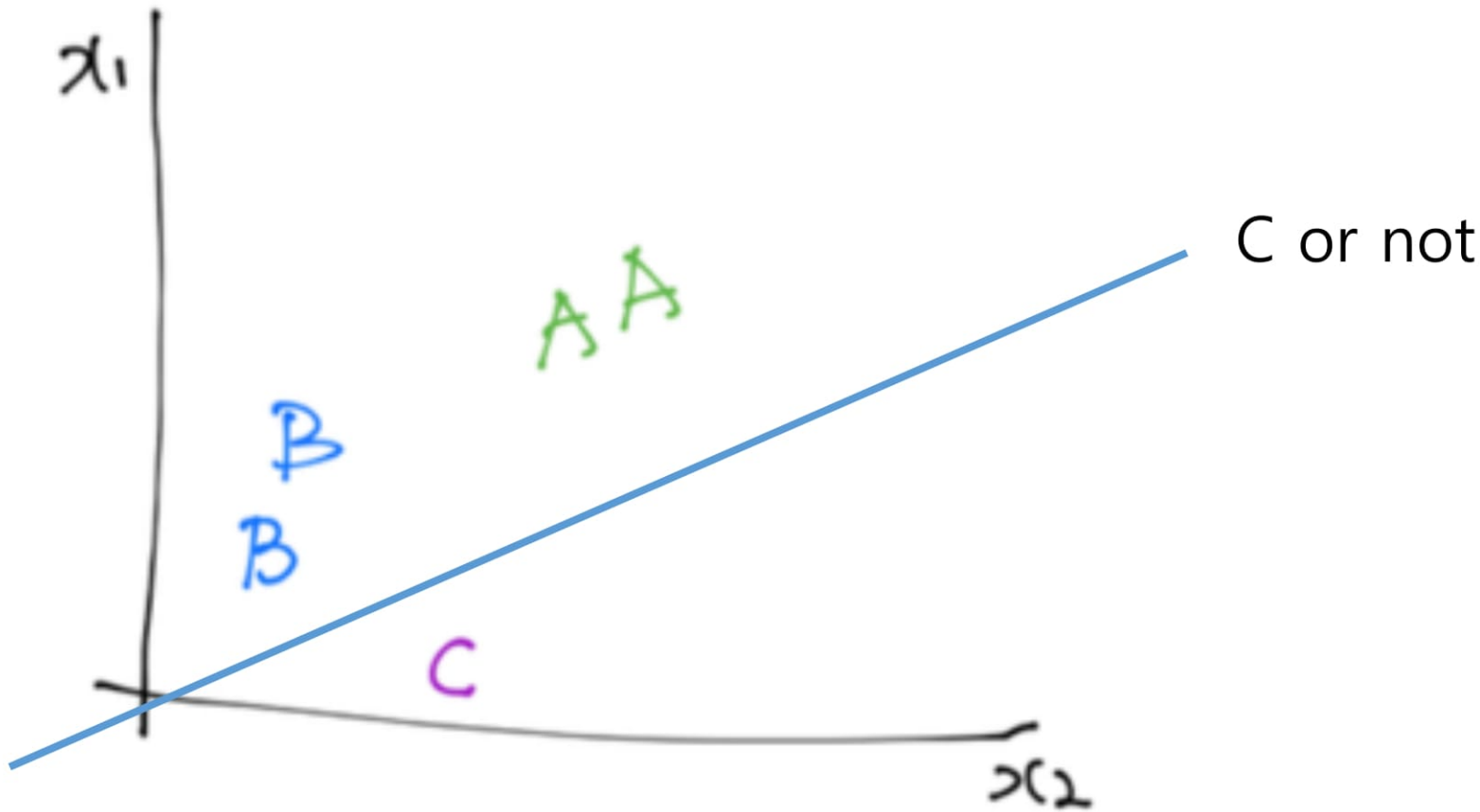
- Letter grade(A, B, C, D, and F) based on time spent
  - **Multi-label classification**

Multinomial classification

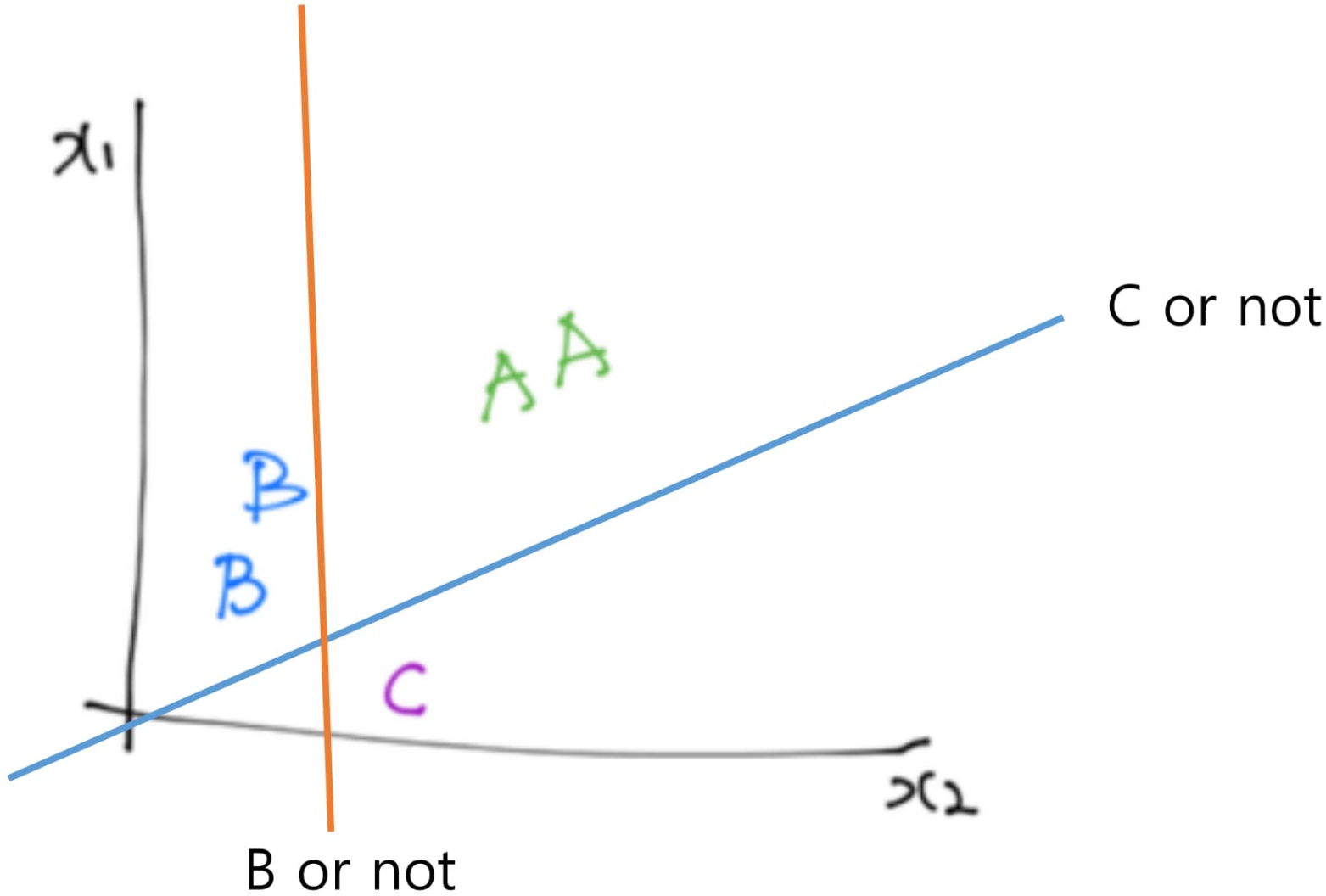
x1 (hours)	x2 (attendance)	y (grade)
10	5	A
9	5	A
3	2	B
2	4	B
11	1	C



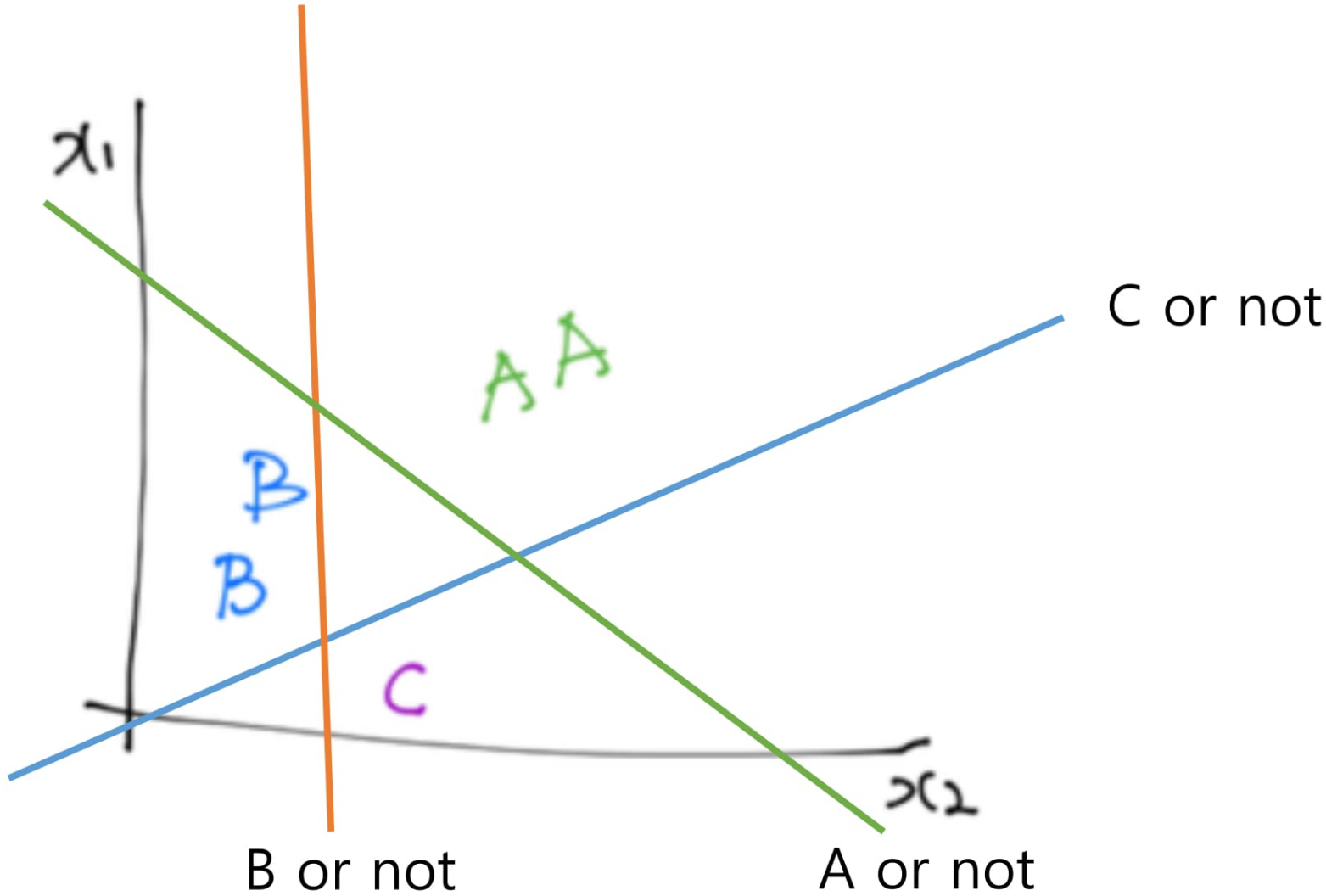
Multinomial classification



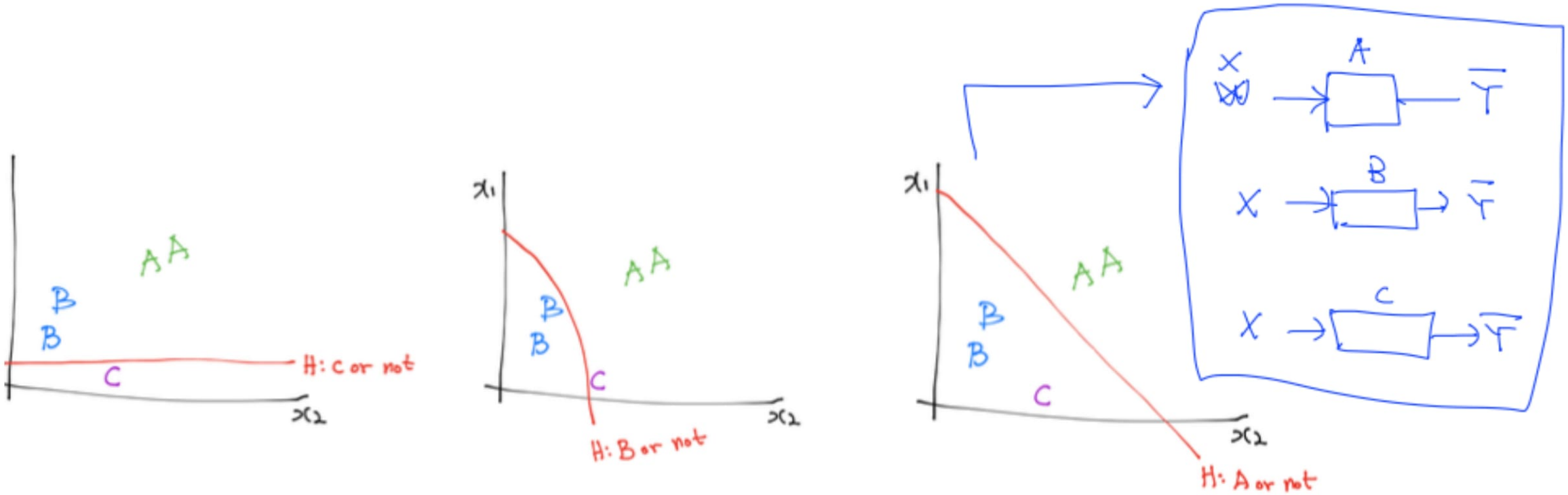
Multinomial classification



Multinomial classification



Multinomial classification



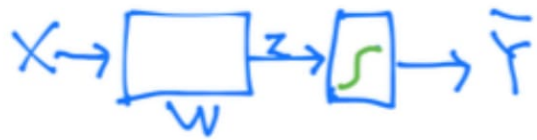
Multinomial classification

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

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

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

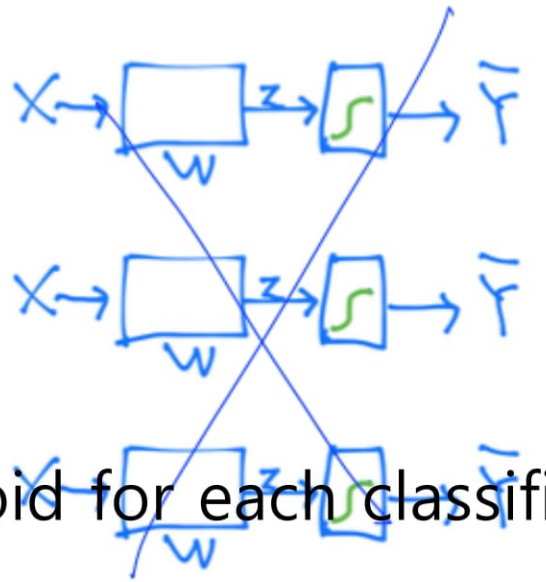
Hypothesis



Activation function

Where is sigmoid?

$$\begin{bmatrix} w_{A1} & w_{A2} & w_{A3} \\ w_{B1} & w_{B2} & w_{B3} \\ w_{C1} & w_{C2} & w_{C3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} w_{A1}x_1 + w_{A2}x_2 + w_{A3}x_3 \\ w_{B1}x_1 + w_{B2}x_2 + w_{B3}x_3 \\ w_{C1}x_1 + w_{C2}x_2 + w_{C3}x_3 \end{bmatrix} = \begin{bmatrix} \overline{y}_A \\ \overline{y}_B \\ \overline{y}_C \end{bmatrix}$$



1. Sigmoid for each classifier is
too much work!

2. Does not add up to 1(inappropriate for probabilities)

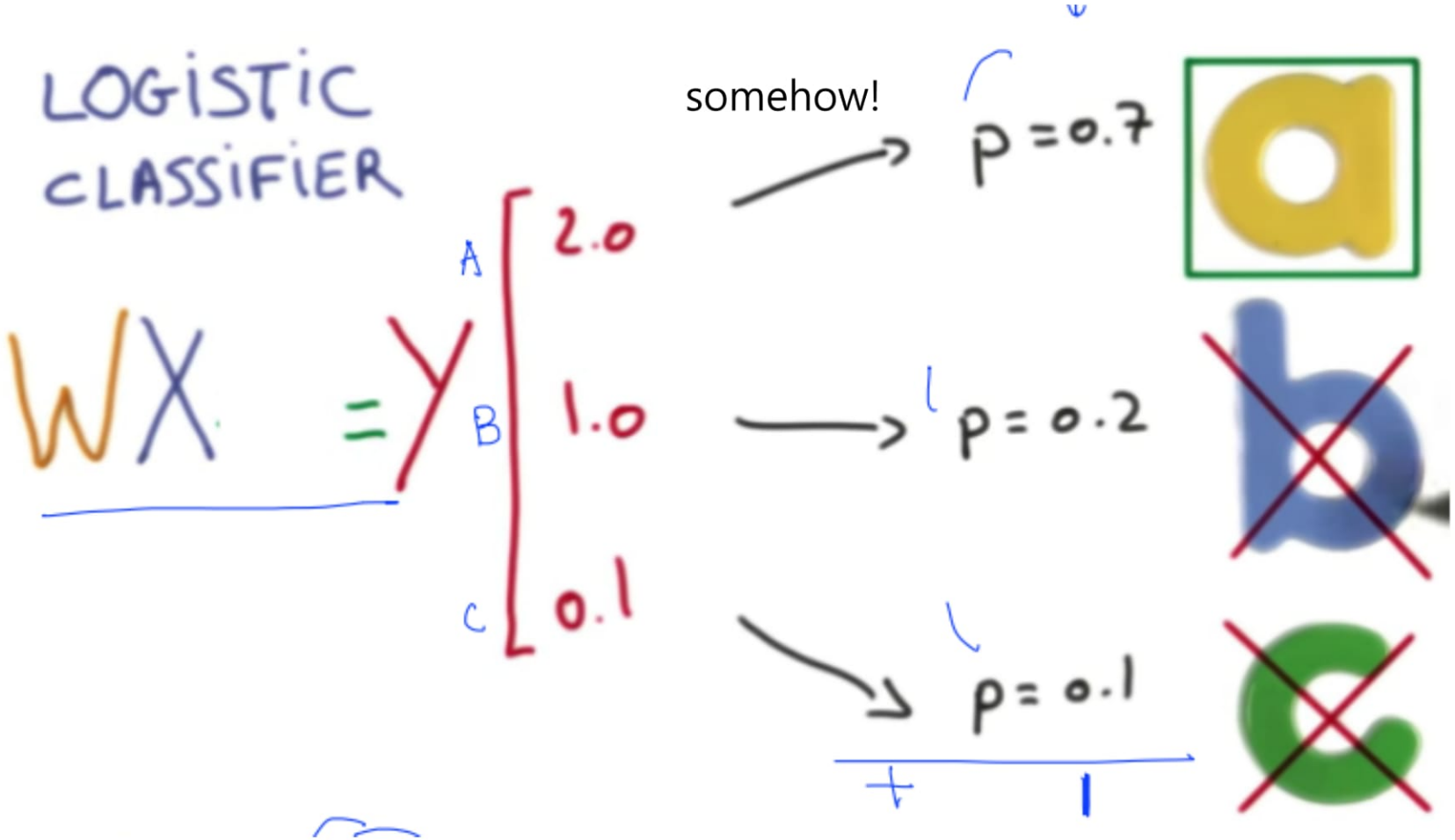
Where is sigmoid?

$$\begin{bmatrix} w_{A1} & w_{A2} & w_{A3} \\ w_{B1} & w_{B2} & w_{B3} \\ w_{C1} & w_{C2} & w_{C3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} w_{A1}x_1 + w_{A2}x_2 + w_{A3}x_3 \\ w_{B1}x_1 + w_{B2}x_2 + w_{B3}x_3 \\ w_{C1}x_1 + w_{C2}x_2 + w_{C3}x_3 \end{bmatrix} = \begin{bmatrix} \overline{y}_A \\ \overline{y}_B \\ \overline{y}_C \end{bmatrix} \begin{bmatrix} 2.0 \\ 1.0 \\ 0.1 \end{bmatrix}$$

0 ~ 1

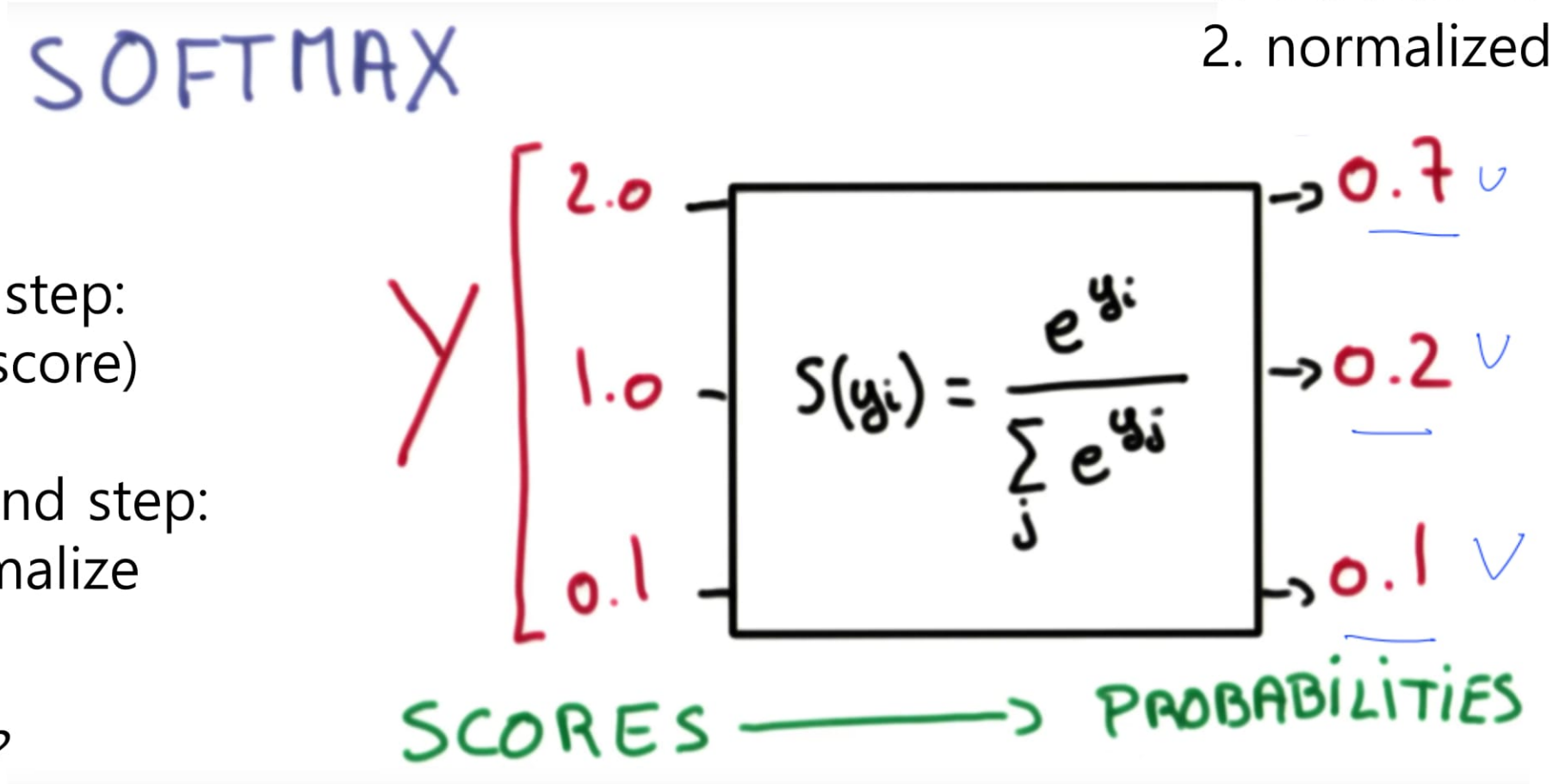
We need an activation function that outputs values between 0 ~ 1, and that add up to 1

No more sigmoid



Softmax regression

Output values are:
1. between 0 ~ 1
2. normalized



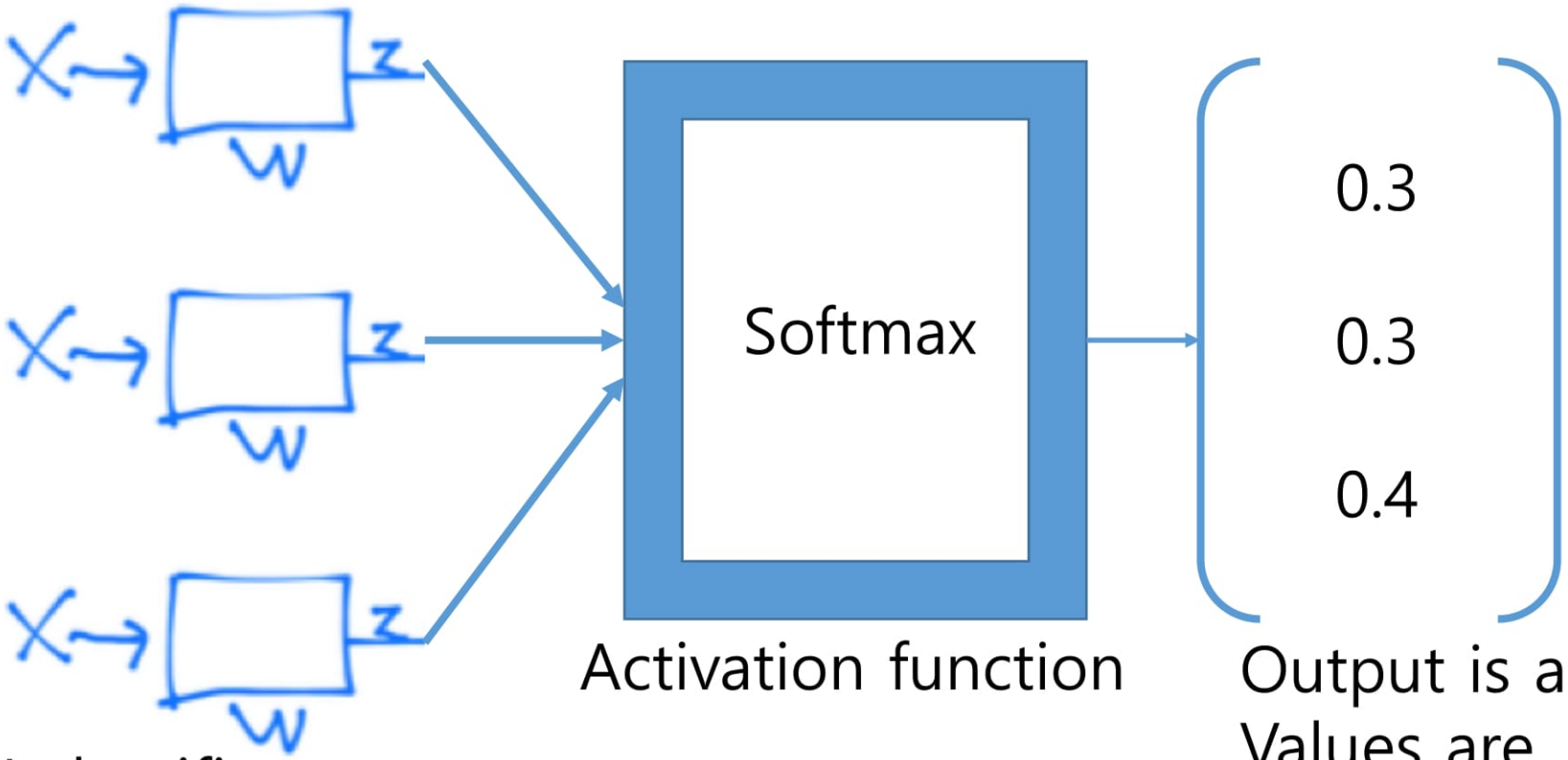
First step:
exp(score)

Second step:
Normalize

Why "e"?

Softmax regression in a nutshell

Dataset



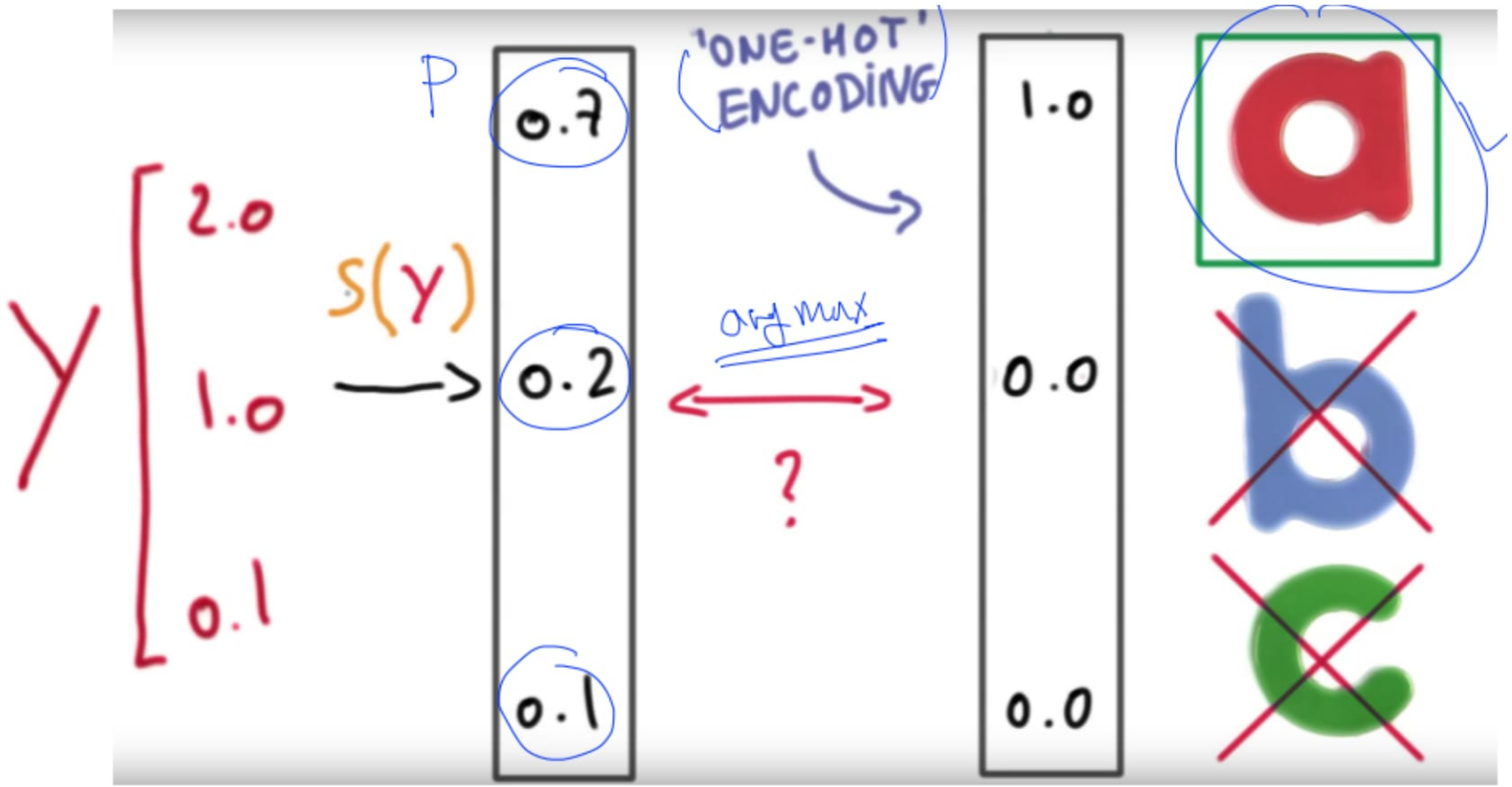
N classifiers for N outputs

Activation function

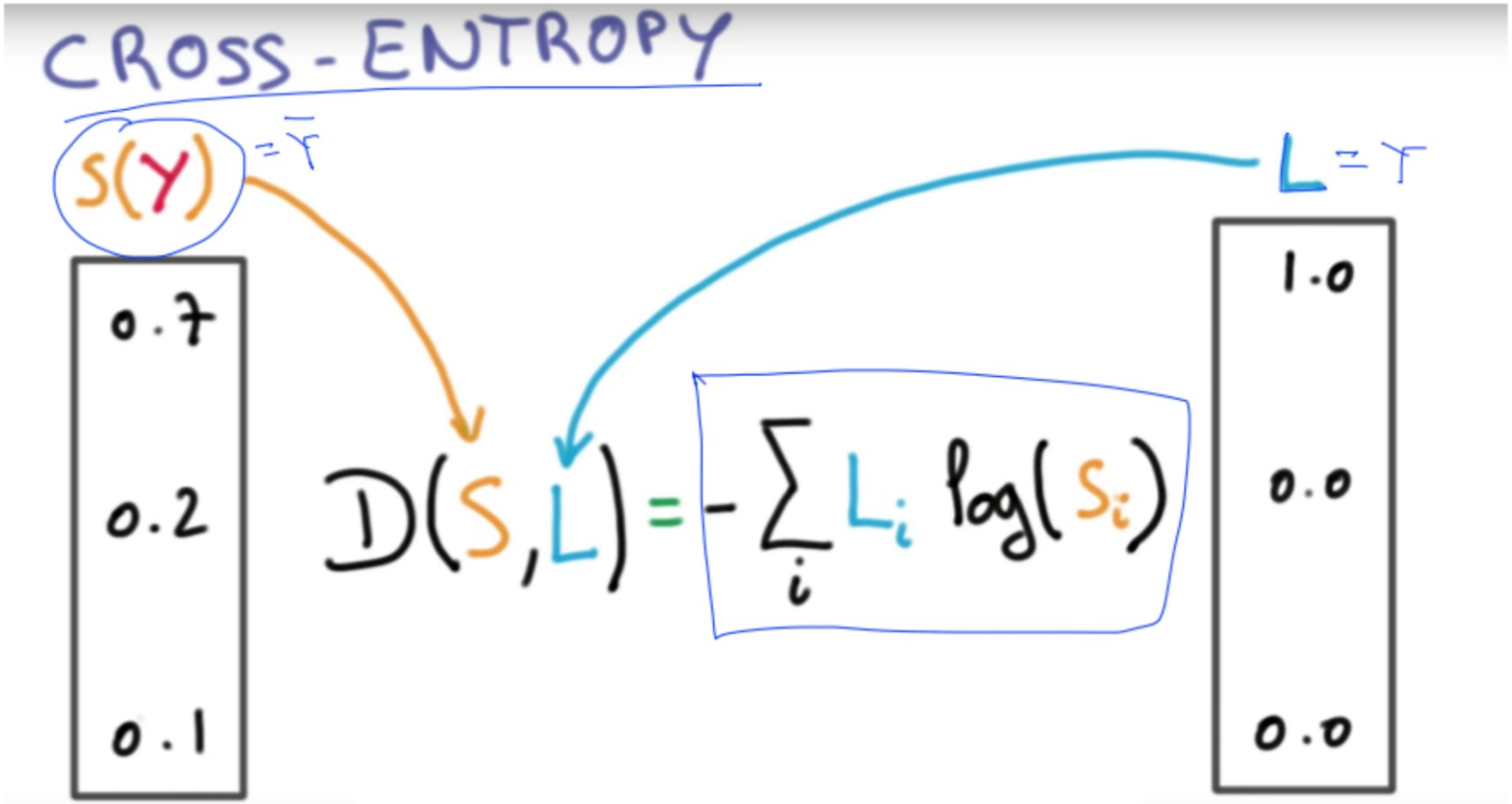
Output is a vector.
Values are

1. between 0 ~ 1
2. normalized

One-hot encoding(optional)



Cost function: cross-entropy



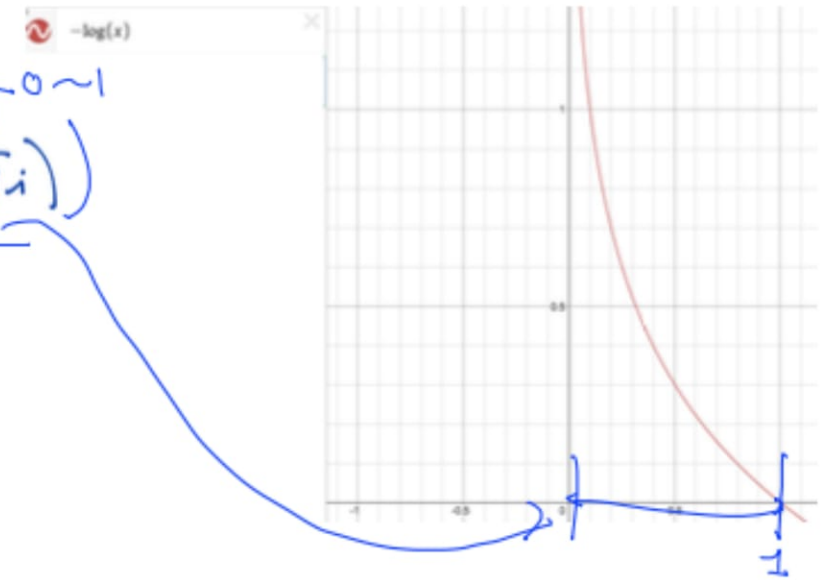
Recall:

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

Real value(answer, label)

Cross-entropy cost function

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

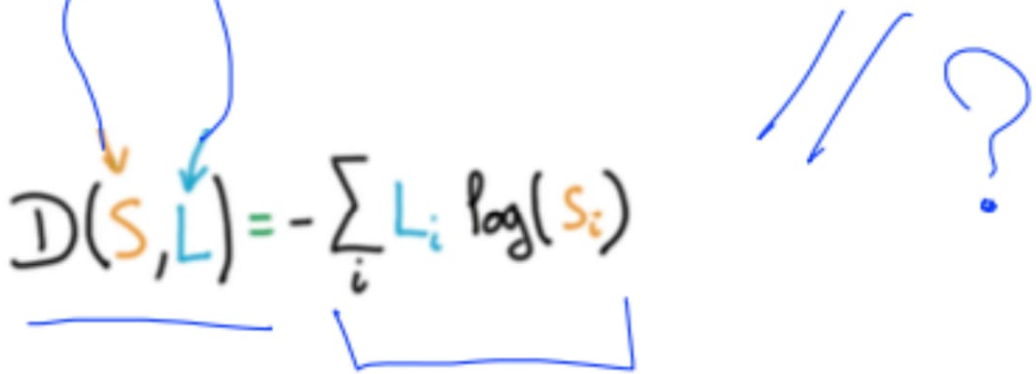


We are only interested in range (0, 1)
why?

Logistic cost vs. cross entropy

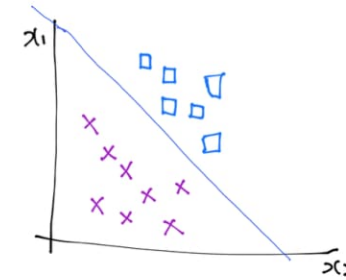
Cost for binary classification

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



Cost for multinomial classification:
generalized logistic cost for n-classification

Training in a nutshell



1

Dataset

Make a model that describes the dataset accurately

H(X)

H(X)

H(X)

Linear model

2

3

Softmax

Activation function

How accurate is the model?

cost

4

Cost function, *cost(W)*

Find W, b that reduces the cost the most

5

Gradient descent

optimizer

6

Update the parameters,
Training done!

So module! Many simple!
Wow.

WOW