

Lecture 5- I

Logistic (regression) classification

Sung Kim <hunkim+mr@gmail.com>

Acknowledgement

- Andrew Ng's ML class
 - <https://class.coursera.org/ml-003/lecture>
 - <http://www.holehouse.org/mlclass/> (note)
- Convolutional Neural Networks for Visual Recognition
 - <http://cs231n.github.io/>
 - <http://cs231n.stanford.edu/>
- TensorFlow
 - <https://www.tensorflow.org>
 - <https://github.com/aymericdamien/TensorFlow-Examples>

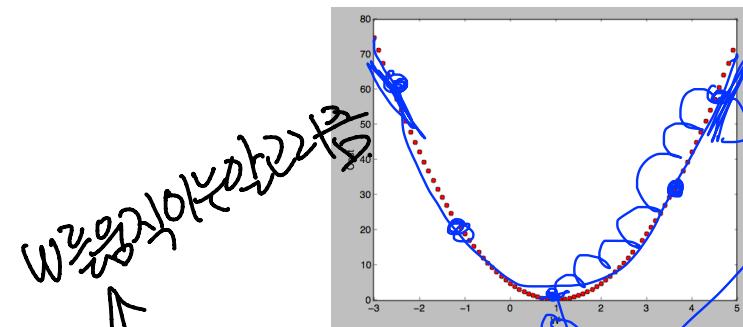
Regression

x_1 (hours)	x_2 (attendance)	y (score)
10	5	90
9	5	80
3	2	50
2	4	60
11	1	40

Gradient w.r.t W minimize $J(W)$
 $\text{cost}(W)$ minimize $J(W)$

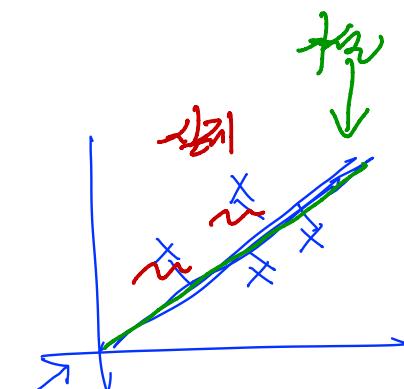
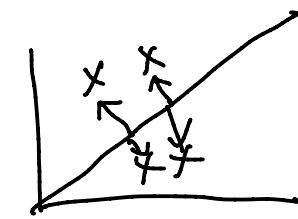
- Hypothesis: $H(X) = WX$

- Cost: $\text{cost}(W) = \frac{1}{m} \sum (WX - y)^2$



- Gradient decent:

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



learning rate
 α (step size or learning rate)
 $\frac{\partial}{\partial W}$ gradient

Binary

Classification

(Spam or Ham) 나는 선생)

Gmail

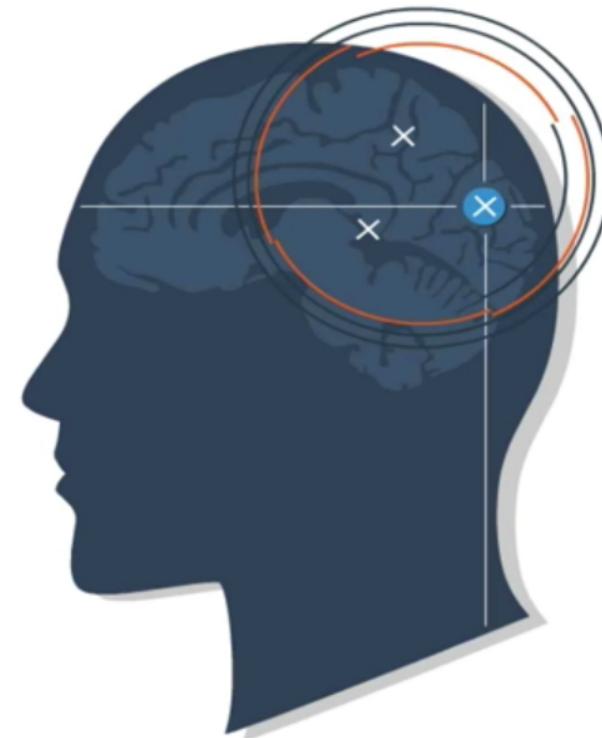
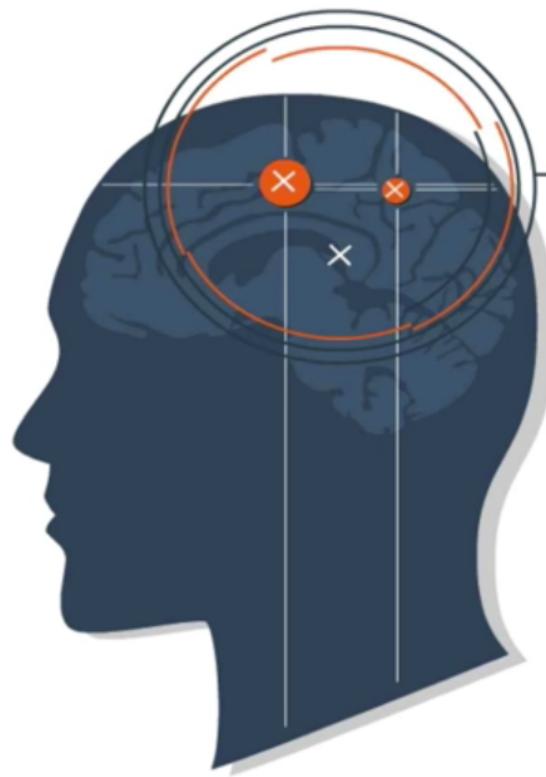
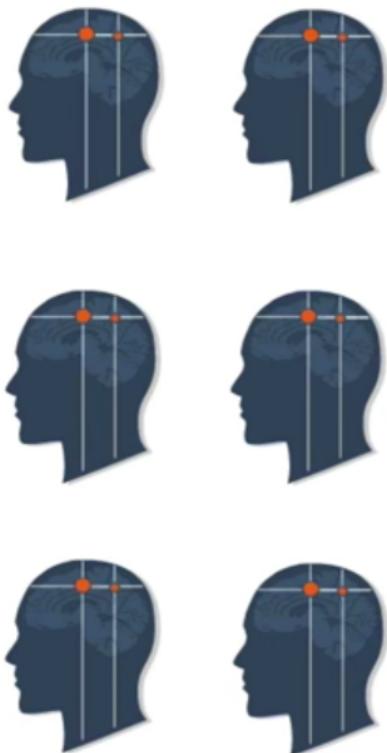
- Spam Detection: Spam or Ham what we like
- Facebook feed: show or hide
- Credit Card Fraudulent Transaction detection: legitimate/fraud



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)

Radiology

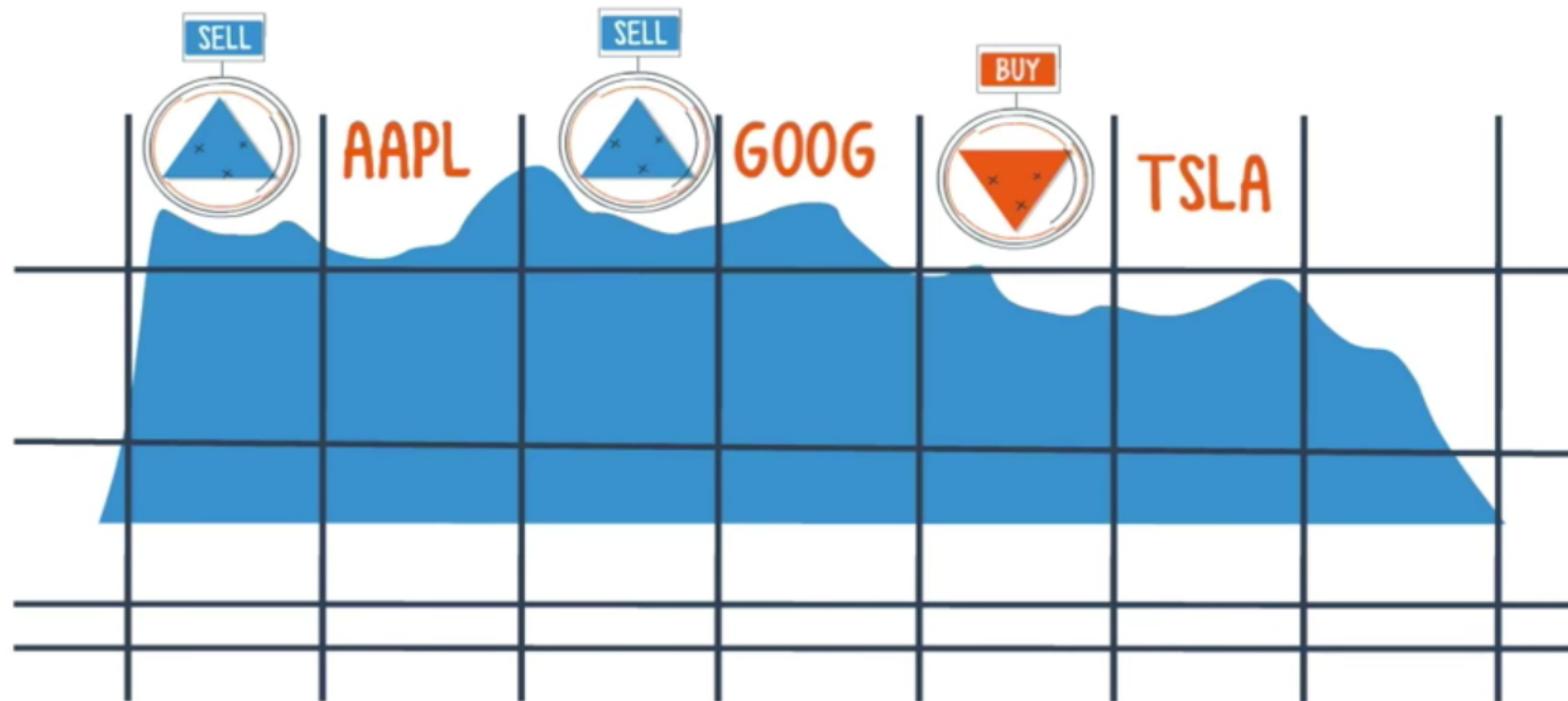


Malignant
tumor

Benign
tumor

Finance

DWJI	17,499.10	▼
SP500	2,025.51	▼
NASDAQ	4,976.9	▲
AAPL	107.71	▲
GOOG	750.06	▲
TSLA	234.24	▼

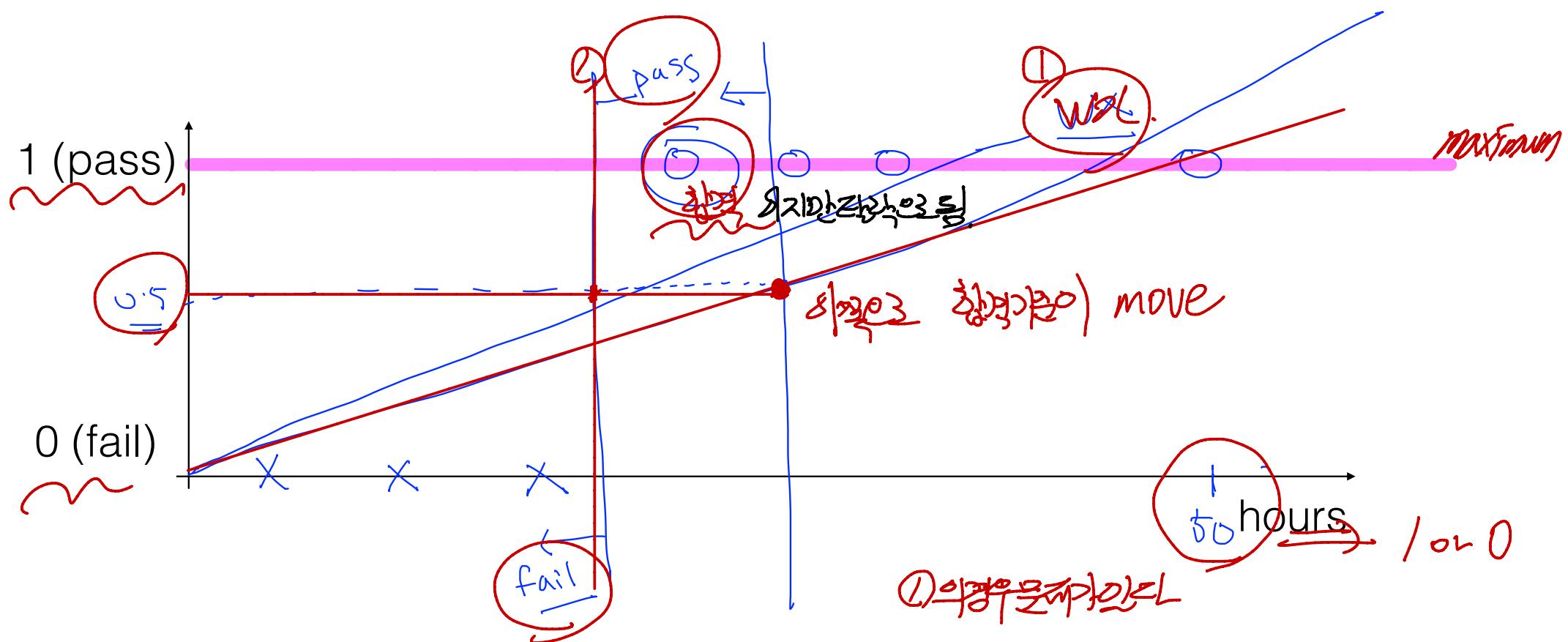


Pass(1)/Fail(0) based on study hours



Linear Regression?

Jimilang



Linear regression

- We know Y is 0 or 1

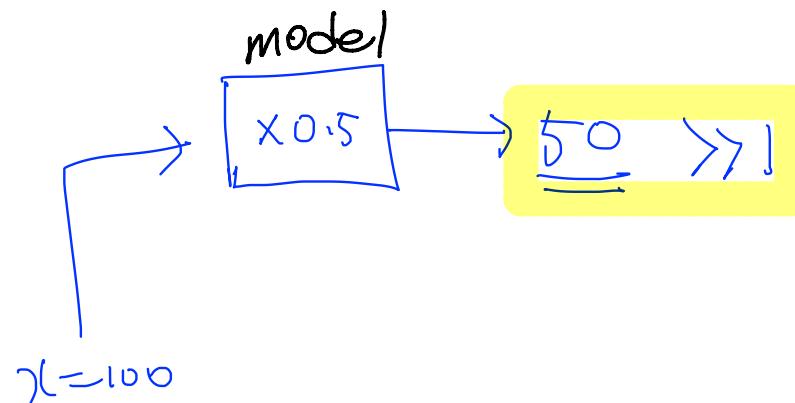
$$H(x) = Wx + b$$

$\frac{2}{\sim 1}$ {
 \nwarrow } \rightarrow 0 or 1.

$$x \begin{bmatrix} 1 \\ 2 \\ 5 \\ 10 \\ 11 \end{bmatrix}, \quad w \sim 0.5, \quad b=0$$

$0 < \sim 1,$

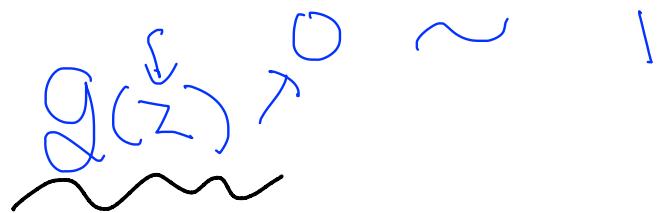
- Hypothesis can give values large than 1 or less than 0

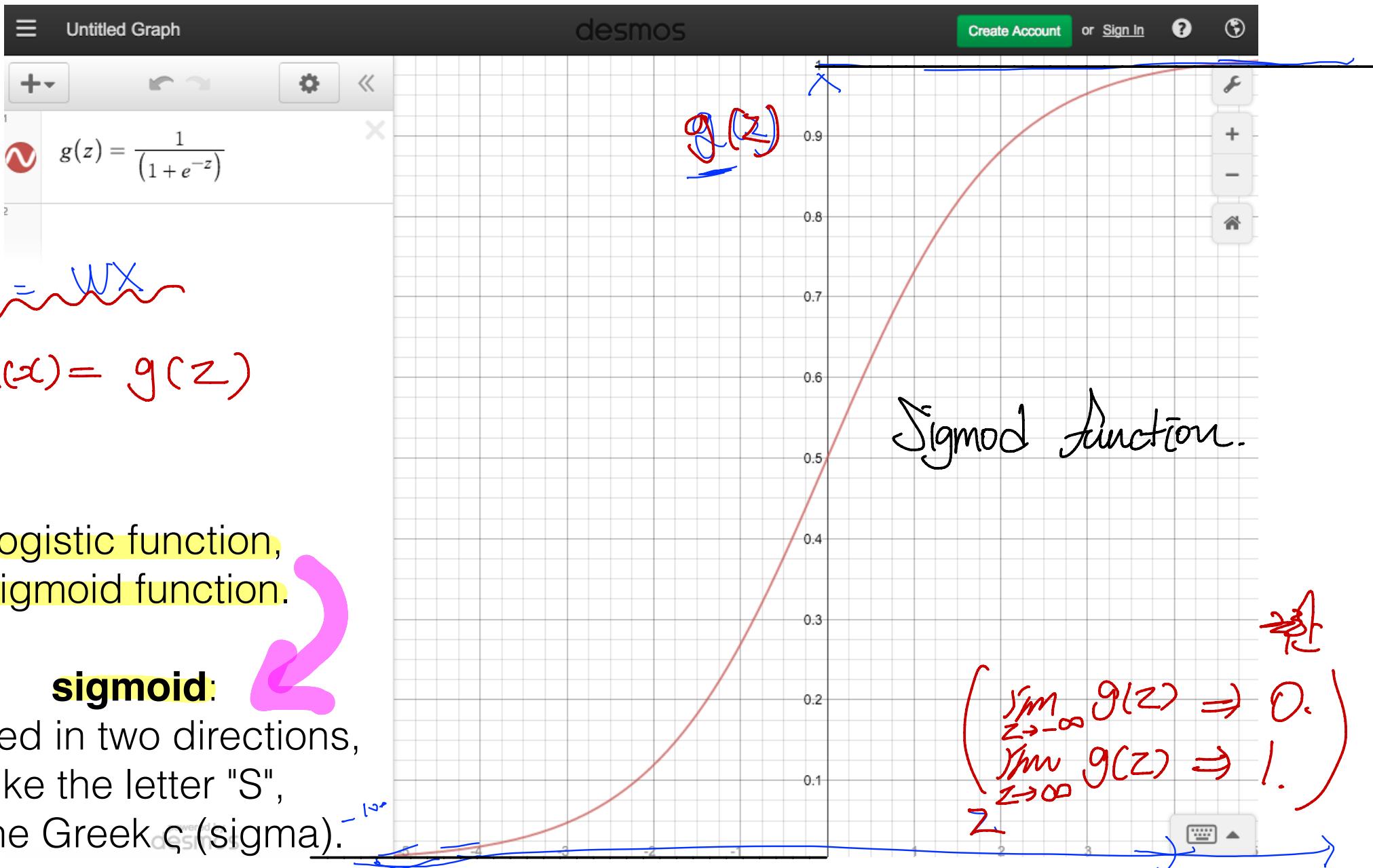


Logistic Hypothesis

$$H(x) = \underline{Wx + b}$$

여기서 주의할 것은
 $H(x) \Rightarrow 0 \sim 1$ 로 압축되는 것.

$$g(z) \sim 1$$




Logistic Hypothesis

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

WX

$$W^T X = Z.$$

Next : Cost & minimize

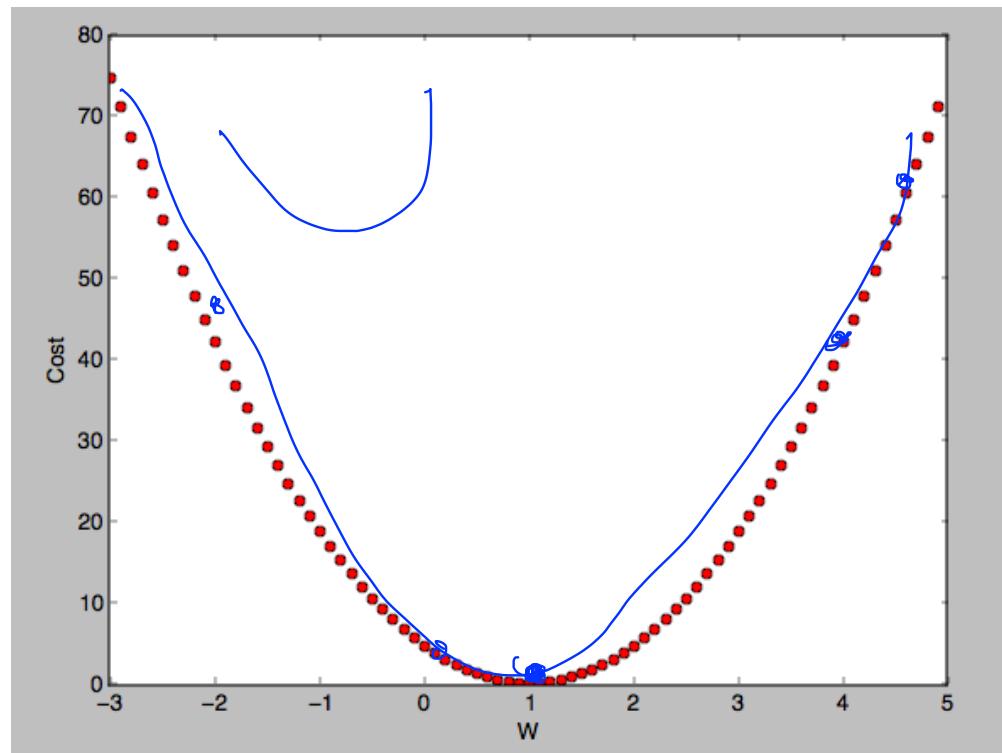
Lecture 5-2

Logistic (regression) classification: cost function & gradient decent

Sung Kim <hunkim+mr@gmail.com>

Cost

$$cost(W, b) = \frac{1}{m} \sum_{i=1}^m (H(x^{(i)}) - y^{(i)})^2 \quad \text{when } \underline{H(x) = Wx + b}$$

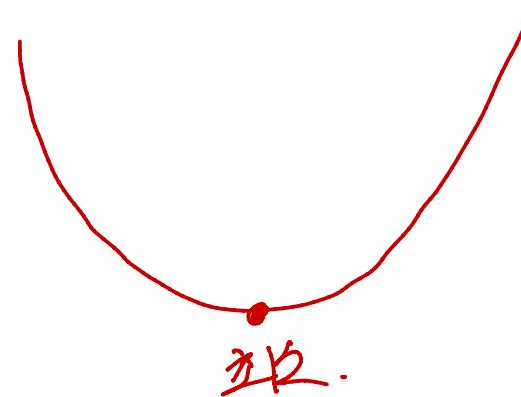


Cost function

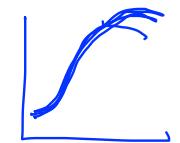
$$cost(W, b) = \frac{1}{m} \sum_{i=1}^m (H(x^{(i)}) - y^{(i)})^2$$

OK ~ < 1

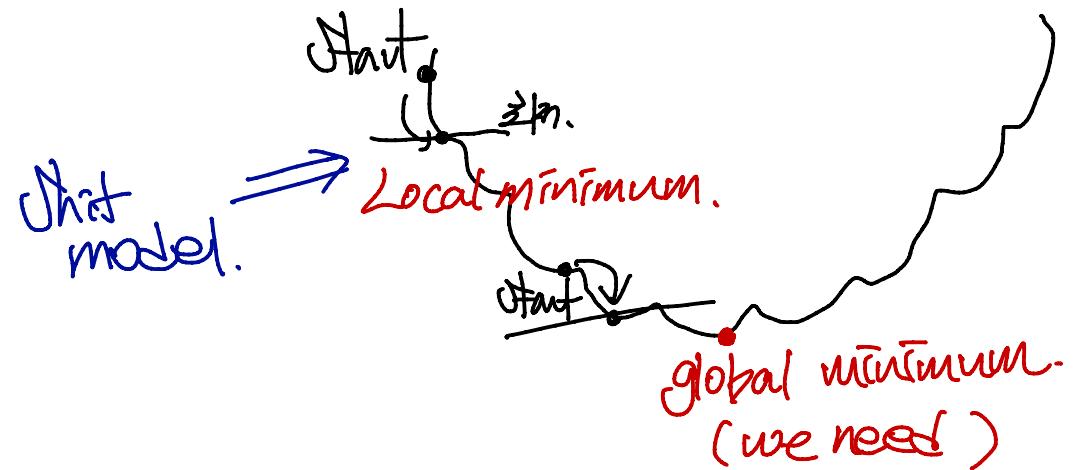
$$H(x) = Wx + b \quad \Rightarrow \quad \text{J}^2$$



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



Not linear.



New cost function for logistic

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

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

Next

understanding cost function

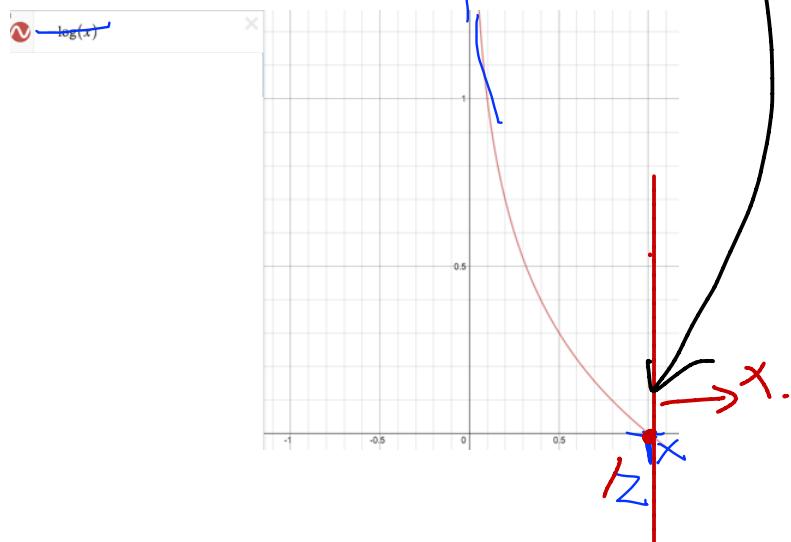
Cost : 예측이 맞으면 ↓ 망이 틀리면 ↑

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

$\frac{1}{1+e^{-x}}$

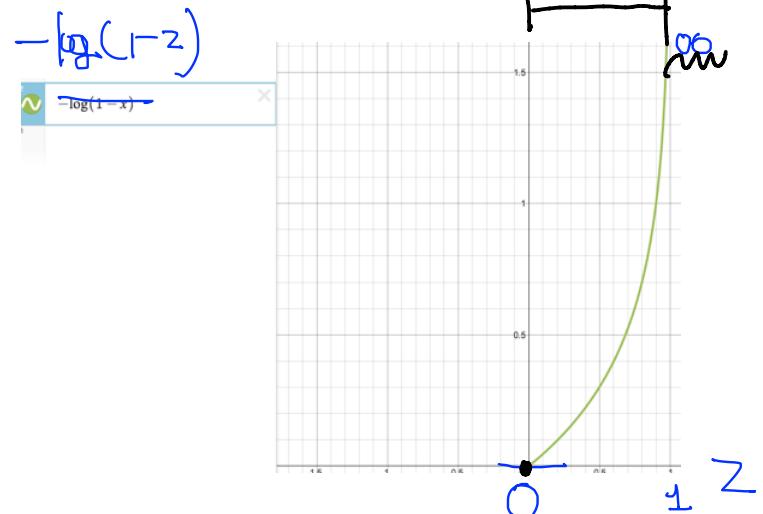
ex) $y=1$ \Rightarrow cost(1) \Rightarrow 0이 됩니다.

$$\text{Cost} \quad g(z) = -\log(z)$$



$y=0$ → 정답

- 1) $H(x) = 0 \quad \text{cost}(0) \Rightarrow 0.$
- 2) $H(x) = 1 \quad \text{cost}(1) \Rightarrow \infty \uparrow$



Cost function

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

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

⇒ programming complexity.

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

$$y=1 \quad (1 - y) \log(1 - H(x)) \Rightarrow \textcircled{1}$$



$$y=1 \quad -y \log(H(x))$$

$$y=0 \quad -y \log(H(x)) \Rightarrow \textcircled{2}$$



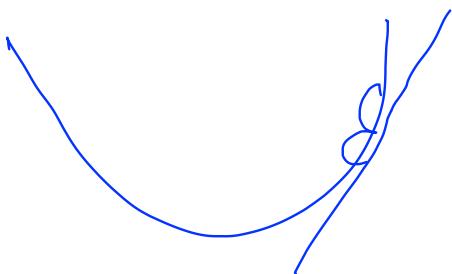
$$y=0 \quad -1 * \log(1 - H(x))$$

Minimize cost - Gradient decent algorithm

경사하강법.

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

경사하강법.



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

→ 광고.

Gradient decent algorithm

$$\underbrace{cost(W) = -\frac{1}{m} \sum y \log(H(x)) + (1-y) \log(1-H(x))}_{W := W - \alpha \frac{\partial}{\partial W} cost(W)}$$

```
# 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)
```

Next
Multinomial
classification (Softmax)

