선형 분류

무인이동체공학과 17011882 김 우 혁

- How to binary classification using linear regression?

∴ [Linear regression → Logistic regression 이용 ⇓ Binary classification ⇓ multi-class classification]

- 복습: 선형 회귀

⇒ $w_1 \times x_1 + w_2 \times x_2 = y$

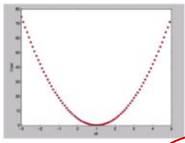
	w1 ~~x1~~ (hours)	w2 ~~x2~~ (attendance)	y (score)
x_1	10	5	90
x_2	9	5	80
x_3	3	2	50
⋮	2	4	60
x_5	11	1	40
	10	3	?

↑ 우리가 예측하고 싶은 것

① 가설 설정
- **Hypothesis:** $H(X) = WX$

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

예측값 (WX), 정답 (y)



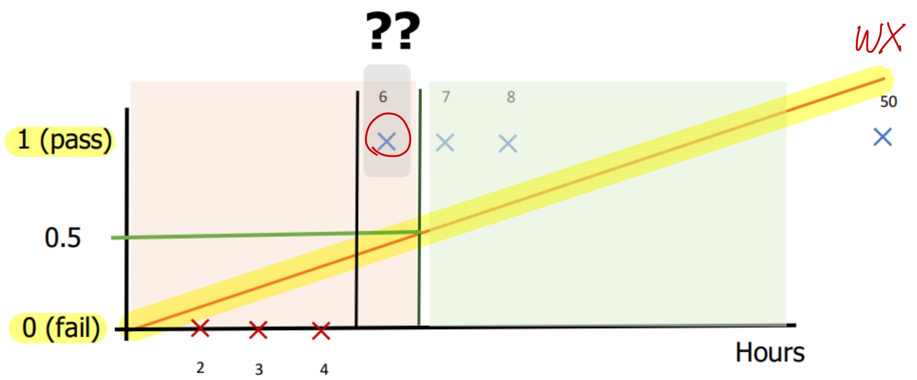
③ 움직이자
- **Gradient decent:** $W := W - \alpha \boxed{\frac{\partial}{\partial W} cost(W)}$

Learning Rate (α)

★ 학습 : 비용 함수값이 최소화되는 W값을 찾는 것

★ 범주형 라벨 → 인코딩 필요.

- **Disadvantage #1** → 선형 회귀 만으로 분류 ⇒ 한계



(정답)
★ 우리는 Y 값이 0 or 1 이라는 사실을 알고 있음.

H(x) = **W**x+**b** —But,→ 1보다 크고 0보다 작은값이 됨...

⬇ **How???**

0<= H(x) <= 1 이렇게 만들고 싶다.

시그모이드 함수 (=로지스틱 함수)

- **Sigmoid:** Curved in two directions, like the letter "S"

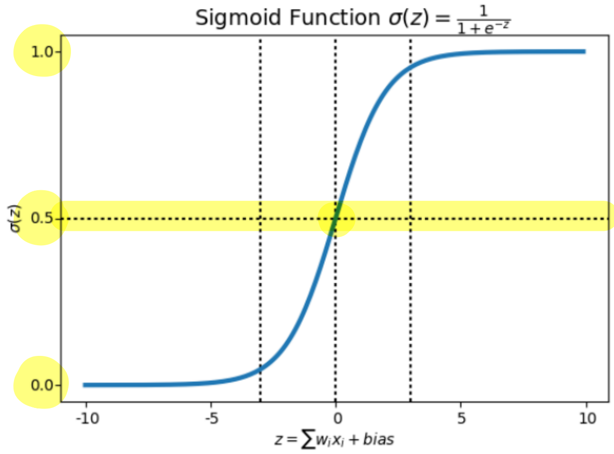
이용

H(X)를 Bound ⇒ 0~1

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

$\boxed{0 \leq H(X) \leq 1}$

Z = WX



Logistic function := sigmoid function

---

① 가설 설정 했으니, 이제 ② Cost function 설정.

Cost function

- 기존 linear regression cost function에 적용하니, local minima에 빠진다.

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

0< ~ <1

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

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

그냥 그려보면...

- 위 그림과 같이 local minima에 빠질 가능성 → cost 재설계 필요!

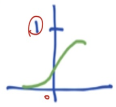
- Cost function: 우리 "예측 값"(가설 값)이 얼마나 "정답"에 가까운가를 측정하는 척도!
  - 정답에 가까워질수록 Cost function값은 작고 정답에서 멀어질수록 Cost function값은 크게! 설계

Cost function for logistic regression

- Logistic regression
  - **H(x) ➔ [0, 1]**
  - if H(x) < 0.5, y = 0 else y = 1
  - 정답에 가까워 질수록 Cost function 값 은 작고, 정답에서 멀어질 수록 Cost function 값은 크게! 설계하면 된다.

$H_L(x) = Wx$ → $0 \le , 1 >$ ∴ 0~1 로 bound!

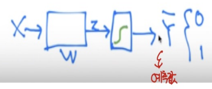
$Z = H_L(x)$, $g(z)$



$g(z) = \frac{1}{1+e^{-z}}$

$H_R(x) = g(H_L(x))$

Rogistic Linear



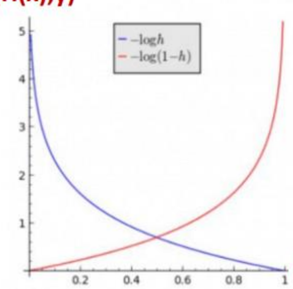
$$H(X) = \frac{1}{1+e^{-W^T X}}$$

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

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

→ 파랑
→ 빨강

C(H(x),y)



★ Gradient decent algorithm

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

$cost(W)$

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

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

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

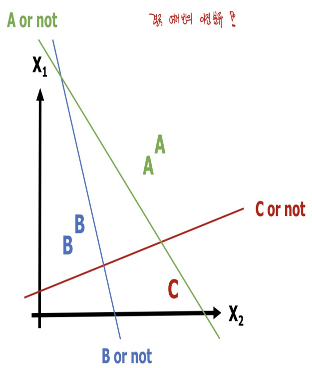
$$c(H(x), y) = \begin{cases} -log(H(x)) & : y = 1 \\ -log(1 - H(x)) & : y = 0 \end{cases}$$

하나의 equation으로 잘 표현하면

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

- How to multi-class classification using binary classification?

Multinomial classification using binary classification



⇓

여러번의 이진분류

= 다중분류

Multinomial classification (여러개의 클래스!)

x1 (hours)	x2 (attendance)	y (grade)
10	5	A 90
9	5	A 80
3	2	B 70
2	4	B 60
11	1	C 40

10 3 ?

w_1 w_2

클래스 3개

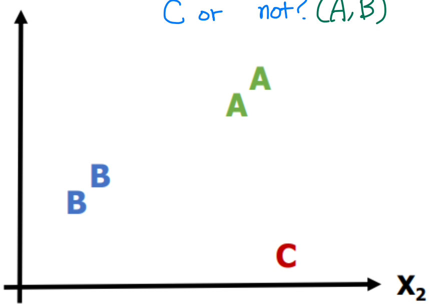
↳이걸 판단할 때.

A or not? (B, C)

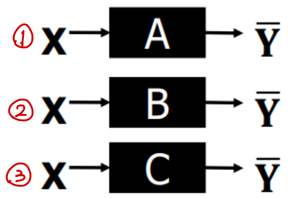
B or not? (C, A)

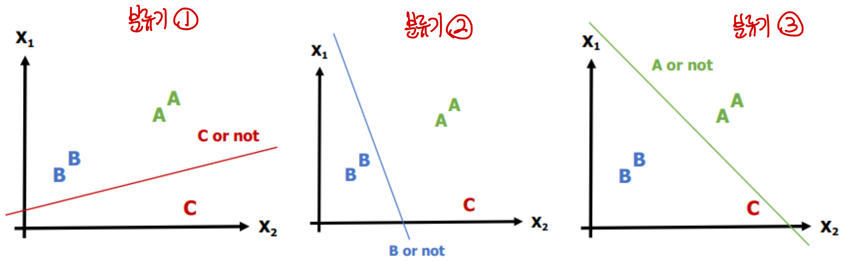
C or not? (A, B)

이런식으로 질문하여 C 라는 결론!

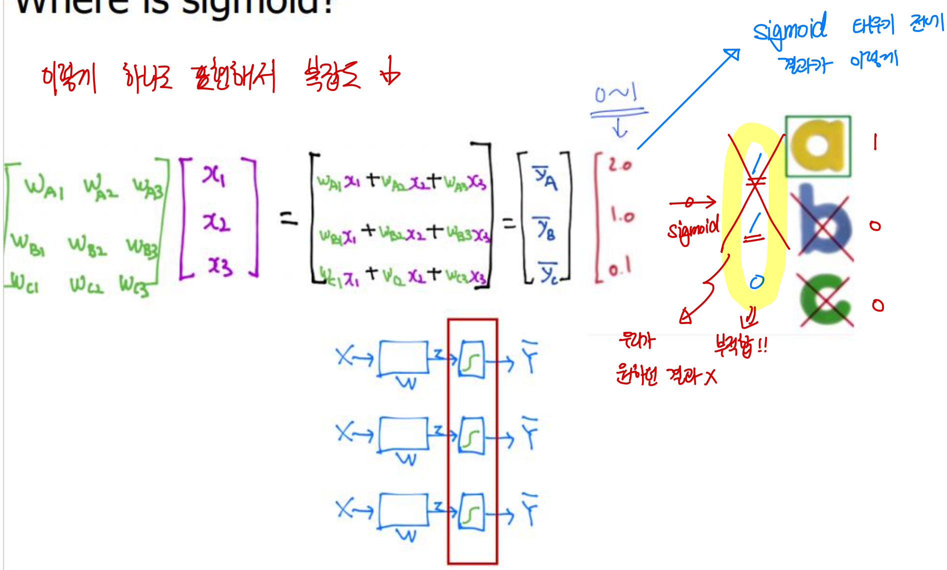


Multinomial classification using binary classification

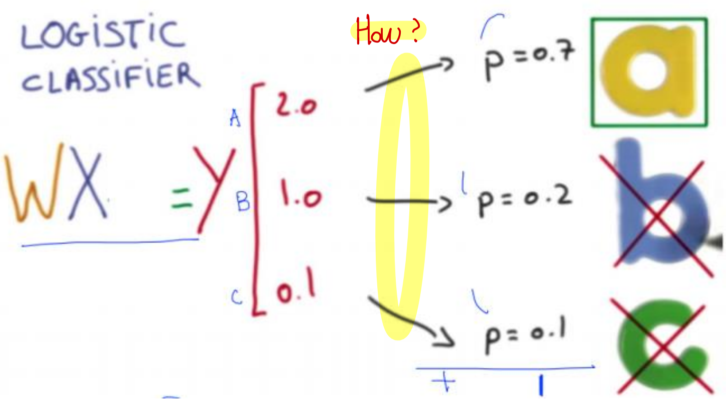




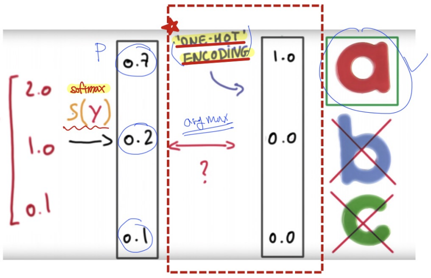
Where is sigmoid?



Multiclass classification의 **Hypothesis**를 **효율적으로 [0,1]** 로 **제한하는 방법**을 알아보자.

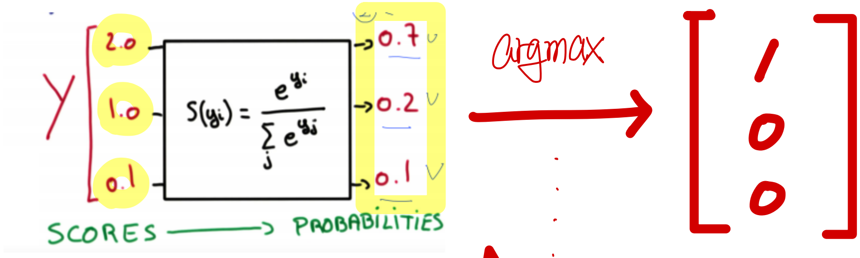


Softmax function (:=Probability)



Softmax function (:= Probability)

1) 모든 값이 0~1 사이
2) 전체 합이 1 (확률 정규화!!)

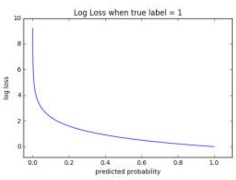


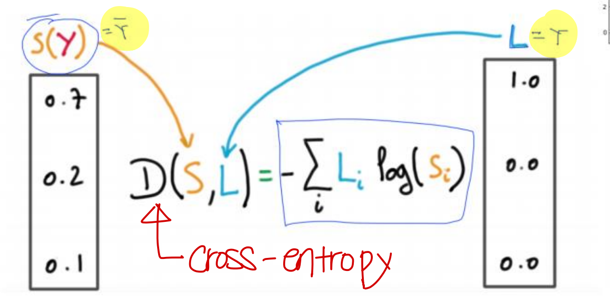
✓ 가설 재설정 했으므로 비용 함수도 다시~

one-hot encoding ⇒ 가장 큰 한가지의 숫자만 1로 변환시키는 방법론

Cost function for multinomial classification

- **Cross-entropy**





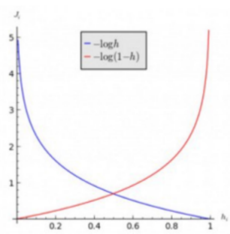
~~Sigmoid~~(Y) softmax

GT: Y

Cross-entropy cost function

predict label

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



element -wise 곱

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

가설이 참 일때 $\bar{Y} = \begin{bmatrix}0\\1\end{bmatrix} = B, \begin{bmatrix}0\\1\end{bmatrix} \odot -log\begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}0\\1\end{bmatrix} \odot \begin{bmatrix}\infty\\0\end{bmatrix} = \begin{bmatrix}0\\0\end{bmatrix}$ => 0

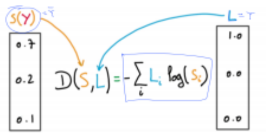
가설이 거짓일 때 $\bar{Y} = \begin{bmatrix}1\\0\end{bmatrix} = A, \begin{bmatrix}0\\1\end{bmatrix} \odot -log\begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}0\\1\end{bmatrix} \odot \begin{bmatrix}0\\\infty\end{bmatrix} = \begin{bmatrix}0\\\infty\end{bmatrix}$ => ∞

Cost function

$$\text{Loss function} = \frac{1}{N}\sum_i \overset{\text{cross-entropy}}{D}\Big(\underbrace{S(WX_i + b)}_{\text{softmax}}, \underbrace{L_i}_{\text{GT}}\Big)$$

$$\mathrm{D(S, L)} = -\sum_i L_i \log(S_i)$$: cross entropy

↳ Loss 구하면 됨



$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}}$$: softmax

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