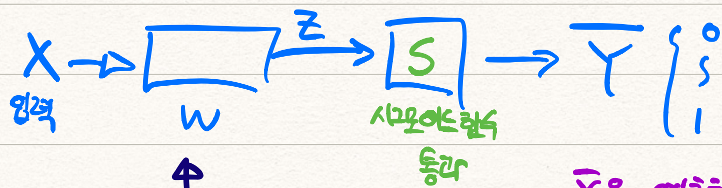
⊛ Logistic regression

$H(x) = W(x)$ — 리턴하는게 실수 [100, 200, ...]

$Z = H_L(x)$, $g(Z)$ ← 얘네를 압축해서 0~1 사이 값을 내어주게끔!.

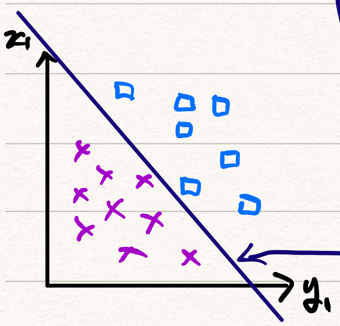
$g(Z) = \frac{1}{1+e^{-z}}$ — '로지스틱', '시그모이드'라고 불림.

$H_R(x) = g(H_L(x))$ ← 최종 hypothesis.



$\bar{Y}$은 예측한 데이터.

Y는 실제 데이터.



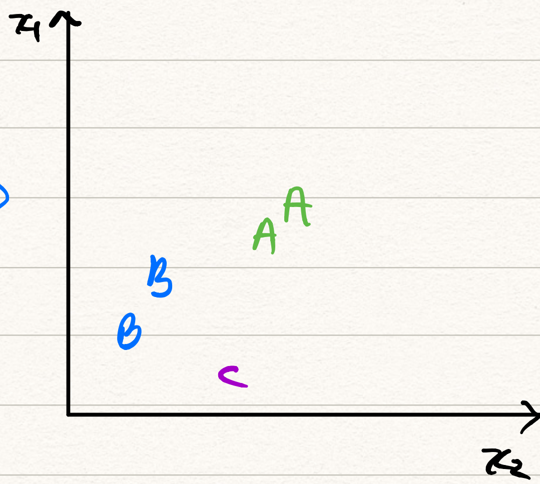
① 저 W를 학습 시킨다

⇒ 데이터들을 구분짓는 '선'을 찾아낸다.

⊛ Multinomial classification

x_1(hours)	x_2(attendance)	y(grade)
10	5	A
9	5	A
3	2	B
2	4	B
11	1	C

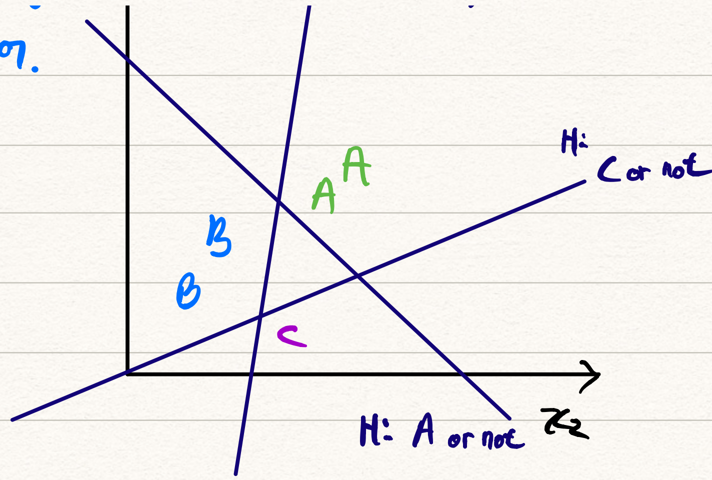
⇒



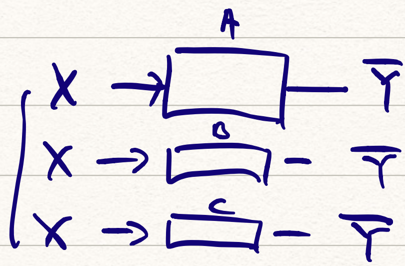
이건 Binary classify 거↑

H: B or not.

로도 할 수 없어.



이때는 각각에 대해 처리해야해 =>



각각에

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

를 구현 해야 함.

→ 독립적으로 구현하기 '복잡하다'

그래서 Metrix를 늘리자.

$$\begin{bmatrix} W_{A1} & W_{A2} & W_{A3} \\ W_{B1} & W_{B2} & W_{B3} \\ W_{C1} & W_{C2} & W_{C3} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} W_{A1}x_1 + W_{A2}x_2 + W_{A3}x_3 \\ W_{B1}x_1 + W_{B2}x_2 + W_{B3}x_3 \\ W_{C1}x_1 + W_{C2}x_2 + W_{C3}x_3 \end{bmatrix} = \begin{array}{c} H_A(x) \\ \begin{bmatrix} y_A \\ y_B \\ y_3 \end{bmatrix} \end{array} \begin{array}{l} \\ \leftarrow H_B(x) \\ \leftarrow H_C(x) \end{array}$$

※ 이제 이 값들을 Sigmoid 로 처리해야해.

cf> Sigmoid.

$$WX = y \begin{cases} 2.0 \longrightarrow P = 0.7 \\ 1.0 \longrightarrow P = 0.2 \end{cases}$$

$\llcorner$ 0.1 ⟶ p = 0.1

/ $\underline{\quad\quad\quad\quad}$

⊕ 1

이 과정에

$\boxed{\text{softmax}}$ 가 핵심.

$$y\begin{bmatrix}2.0\\1.0\\0.1\end{bmatrix} \rightarrow \boxed{S(y_i)=\frac{e^{y_i}}{\sum_j e^{y_j}}} \rightarrow \begin{matrix}0.7\\0.2\\0.1\end{matrix}$$

scores ⟶ probabilities

$$y\begin{bmatrix}2.0\\1.0\\0.1\end{bmatrix} \xrightarrow{S(y)} \begin{bmatrix}0.7\\0.2\\0.1\end{bmatrix} \xleftrightarrow{\text{'one-hot' encoding}} \begin{bmatrix}1.0\\0.0\\0.0\end{bmatrix}$$

a 선택

⊛ Cost function 설계.

$\boxed{\text{Cross-Entropy}}$ 를 사용.

$S(y)$ — softmax 통해 얻은 $\bar{Y}$

$L = Y$ (실제 값)

$$\begin{bmatrix}0.7\\0.2\\0.1\end{bmatrix} \quad D(S, L) = \boxed{-\sum_i L_i \log(S_i)} \quad \begin{bmatrix}1.0\\0.0\\0.0\end{bmatrix}$$

Cross-entropy

$$= -\sum_i L_i \log(\bar{y}_i) = \sum_i (L_i) * \underbrace{(-\log(\bar{y}_i))}_{0\sim1}$$

<ex>

$Y = L = \begin{bmatrix}0\\1\end{bmatrix}$ ⇒ B가 값이 걸리

1) $\bar{Y} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ 예측이 B를 예측한 경우. (OK)

2) $\bar{Y} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ 〃 A를 〃 (X)

→ cost function에는 어차피 예측이 맞는지 여부를 판단.

1) $\begin{bmatrix} 0 \\ 1 \end{bmatrix} \odot -\log\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \odot \begin{bmatrix} \infty \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow 0$

↳ cost가 0이므로 좋은것!

2) $\begin{bmatrix} 0 \\ 1 \end{bmatrix} \odot -\log\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \odot \begin{bmatrix} 0 \\ \infty \end{bmatrix} = \begin{bmatrix} 0 \\ \infty \end{bmatrix} \Rightarrow \infty$

↳ cost가 ↑. 틀린것.

여기서 잠깐!

⊛ Logistic cost VS cross entropy

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

요게 사실 cross entropy였어.

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

? 왜 이게 같은가?

⊛ Cost function

$$\mathcal{L} = \frac{1}{n}\sum_i D(S(WX_i + b), L_i)$$

Loss

Training set.

⊗ Cost 최소화 위해... Gradient descent.

늘 그랬듯이 '미분' 시키자.