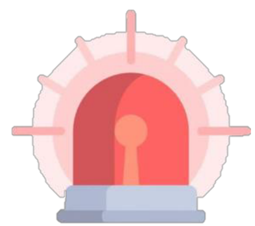
Logistic Regression

Caution

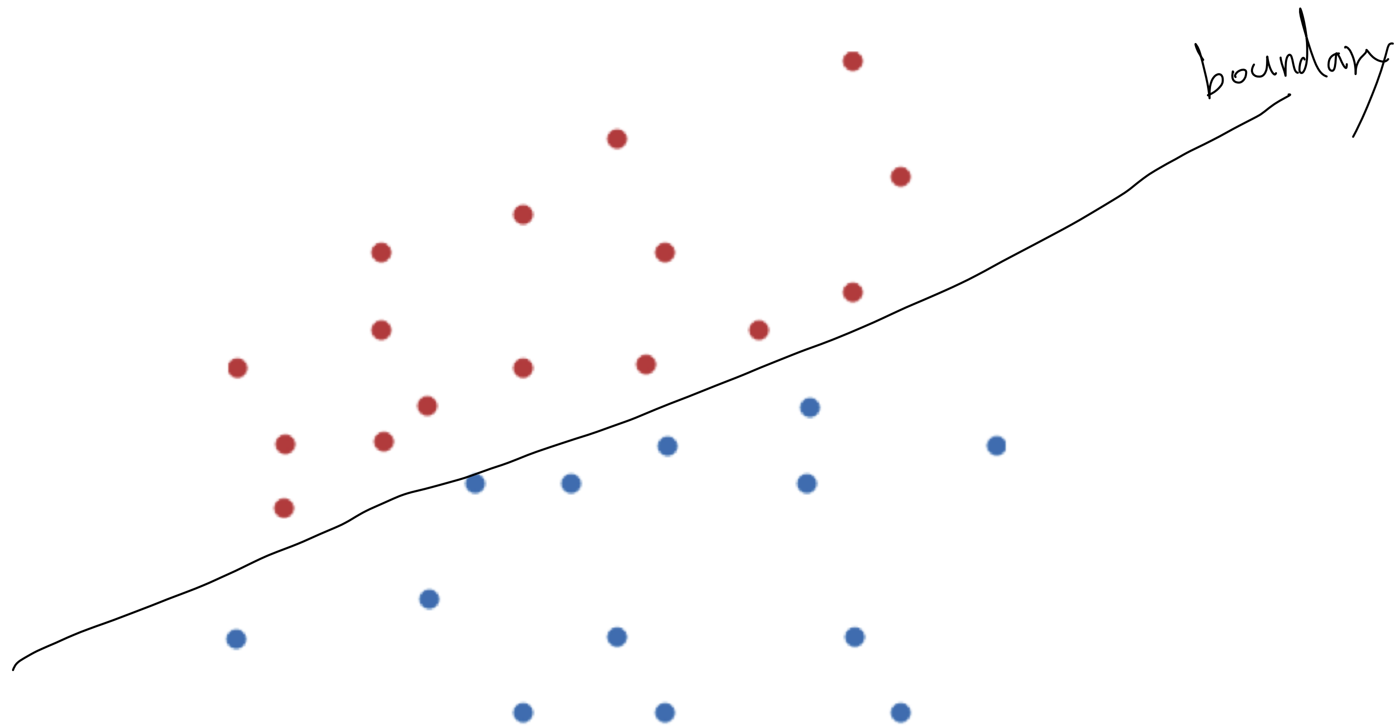


Complex: Differentiation and Probability

One of difficult parts in this class, but
it will be a foundation of Deep Neural Networks!!

Linear Classifier

- **A linear boundary between classes**



Linear Regression
① Define Error fun
② Find w's that minimize E

머신러닝의 기초

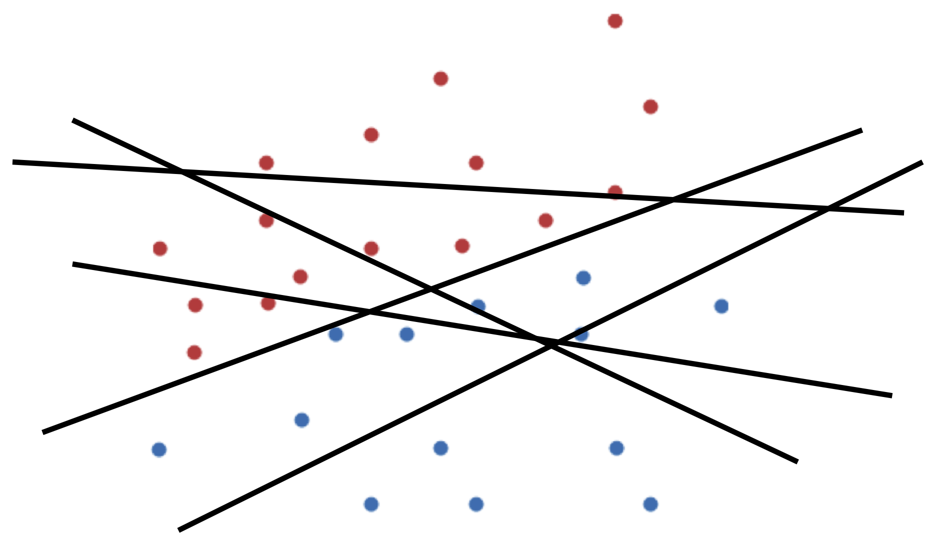
Linear Classifier

똑같다

unknown
양의 음의 정의
어떻게 정답을 내리는지

- **Two things we have to do**
  1. To find the boundary between Red and Blue
  2. Given boundary, To design a classifier, $L(x_1, x_2)$, such that
     - If (x_1, x_2) is above the boundary, $L(x_1, x_2) = Red$
     - If (x_1, x_2) is on the boundary, $L(x_1, x_2) = Unknown$
     - If (x_1, x_2) is below the boundary, $L(x_1, x_2) = Blue$

☆

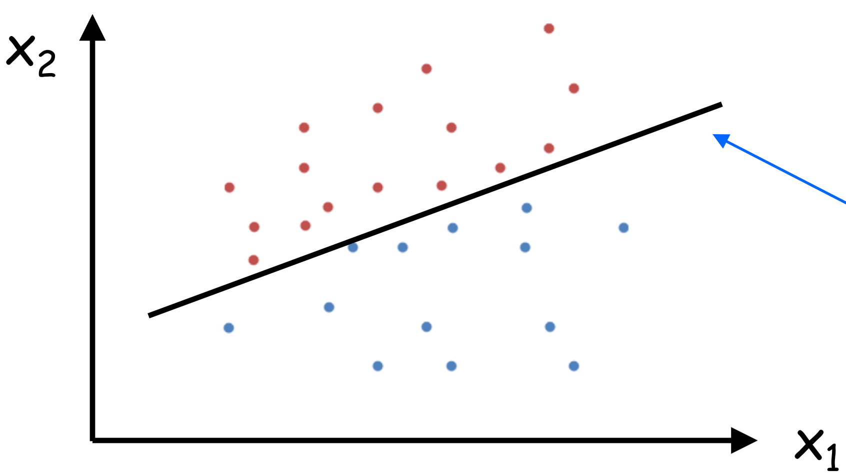


Linear Classifier

- **Classifier Design**
  - If (x_1, x_2) is above the boundary, $L(x_1, x_2) = Red$
  - If (x_1, x_2) is on the boundary, $L(x_1, x_2) = Unknown$
  - If (x_1, x_2) is below the boundary, $L(x_1, x_2) = Blue$

Classfier

↳ 계산을 위해선 공식화해야함
(formular)



$$f(x_1, x_2) = w_0 + w_1 x_1 + w_2 x_2$$

Boundary is given $(w_2 > 0)$

Linear Classifier

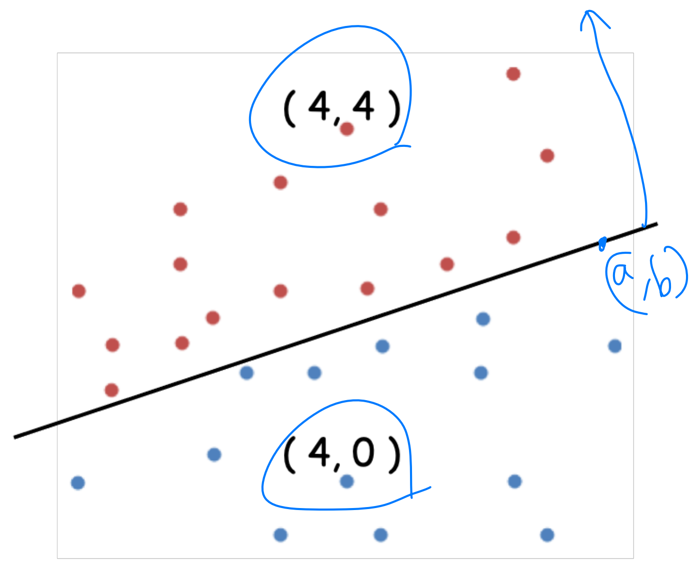
- **Classifier Design**
  - Above the line? Blow the line?

$$f(x_1, x_2) = w_0 + w_1 x_1 + w_2 x_2 \quad (w_2 > 0)$$

$$4w_1 + 4w_2 + w_3 > 0$$
$$4w_1 + 4w_2 + w_3 = 0$$
$$4w_1 + 4w_2 + w_3 < 0$$

Which is true?

$$4w_1 + 0w_2 + w_3 > 0$$
$$4w_1 + 0w_2 + w_3 = 0$$
$$4w_1 + 0w_2 + w_3 < 0$$



f(a,b)=0

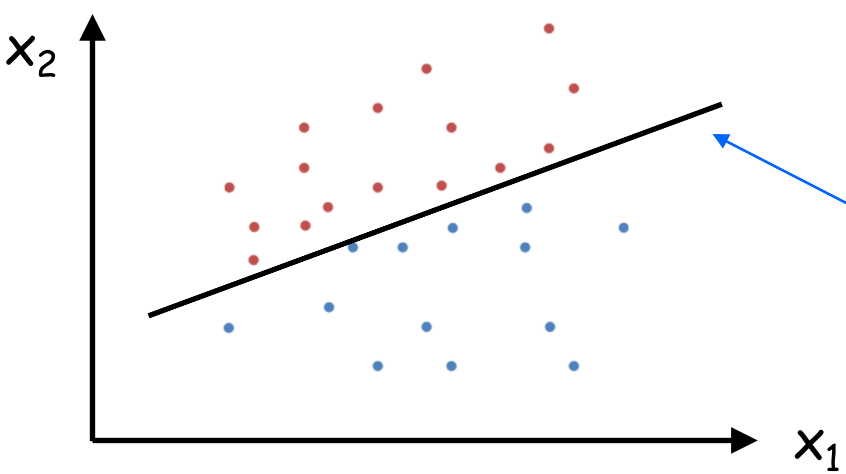
Linear Classifier

▪ Classifier Design

- If (x_1, x_2) is above the boundary, $L(x_1, x_2) = Red$
- If (x_1, x_2) is on the boundary, $L(x_1, x_2) = Unknown$
- If (x_1, x_2) is below the boundary, $L(x_1, x_2) = Blue$

$$\longrightarrow \quad L(x_1, x_2) = \begin{cases} \text{Red} & \text{if } f(x_1, x_2) > 0 \\ \text{Unknown} & \text{if } f(x_1, x_2) = 0 \\ \text{Blue} & \text{if } f(x_1, x_2) < 0 \end{cases}$$

실제 숫자X



Not bad.. But
the output is not real number.
It is a bit inconvenient !

$$f(x_1, x_2) = w_0 + w_1 x_1 + w_2 x_2$$

Boundary is given $(w_2 > 0)$

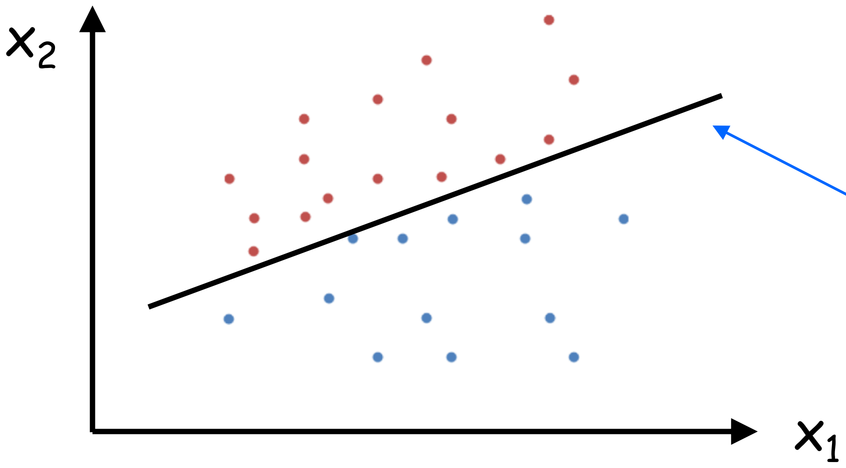
Linear Classifier

▪ Classifier Design

$$L(x_1, x_2) = \begin{cases} \text{Red} & \text{if } f(x_1, x_2) > 0 \\ \text{Unknown} & \text{if } f(x_1, x_2) = 0 \\ \text{Blue} & \text{if } f(x_1, x_2) < 0 \end{cases} \quad \longrightarrow \quad L(x_1, x_2) = \begin{cases} 1 & \text{if } f(x_1, x_2) > 0 \\ 0.5 & \text{if } f(x_1, x_2) = 0 \\ 0 & \text{if } f(x_1, x_2) < 0 \end{cases}$$

연속적이지 X

Hmm.. L is not continuous !!
It is quite inconvenient.

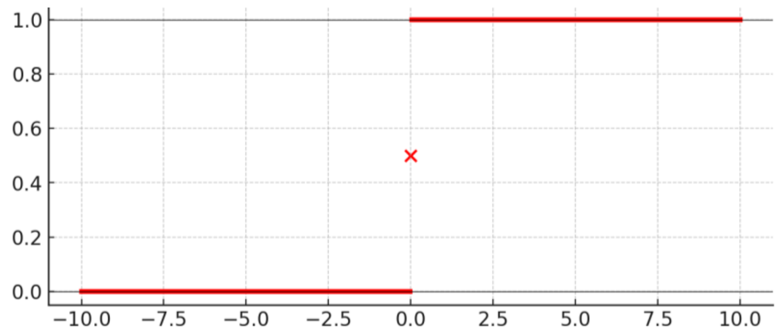


$$f(x_1, x_2) = w_0 + w_1 x_1 + w_2 x_2$$

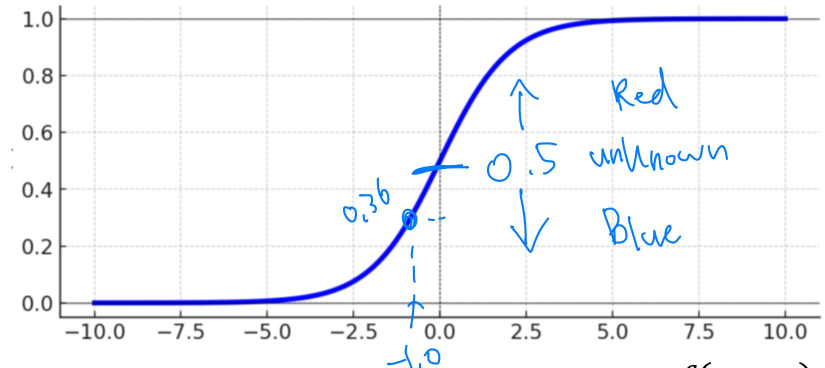
Boundary is given $(w_2 > 0)$

Linear Classifier

- **Classifier Design: Approximation of L**



$f(x_1, x_2)$



$f(x_1, x_2)$

Given Boundary

연속X → 미분 X → GDM 이용못함

Given Boundary

$$L(x_1, x_2) = \begin{cases} 1 & \text{if } f(x_1, x_2) > 0 \\ 0.5 & \text{if } f(x_1, x_2) = 0 \\ 0 & \text{if } f(x_1, x_2) < 0 \end{cases}$$

$$\boxed{L(x_1, x_2) = \frac{1}{1 + \exp^{-f(x_1, x_2)}}}$$

Binary classification만 가능하다. ← Logistic Regression (수업에서 배우는 내용은)

Logistic Regression

이름만 회귀 내용은 classifier

Linear Classifier

이미지를 주고 고양이/강아지인지 판별하는 경우 이미지 1024 1024 10^6

- **Classifier Design: Approximation of L** 차원성이 10^6이 된다
  - Numerical Example 0 < ( ) < 1 ⇒ 확률과 비슷하다

도출하려는 (내가 원하는 값 바)

$$L(x_1, x_2) = \frac{1}{1 + \exp^{-f(x_1, x_2)}}$$

타겟벨류

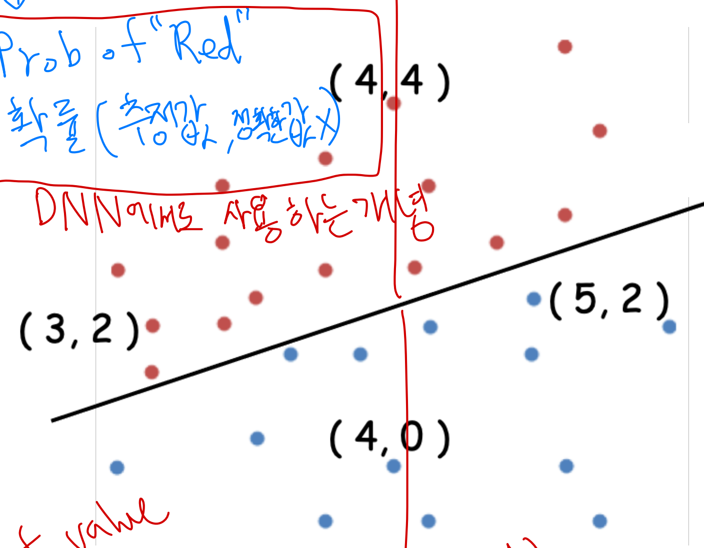
바운더리에서 떨어진정도

regard

$$f(x_1, x_2) = -x_1 + x_2 + 2$$

	Label	t	$f(x_1, x_2)$	$L(x_1, x_2)$
(4, 10)	Red	1	8	0.9997
(4, 4)	Red	1	2	0.8808
(3, 2)	Red	1	1	0.7311
(4, 2)	?	?	0	0.5000
(4, 0)	Blue	0	-2	0.1192
(5, 2)	Blue	0	-3	0.0474
(4,-6)	Blue	0	-8	0.0003

Prob of "Red"
확률 (추정값, 정확한값X)
DNN에서 사용하는개념



approximation

close to target value 인지 확인!

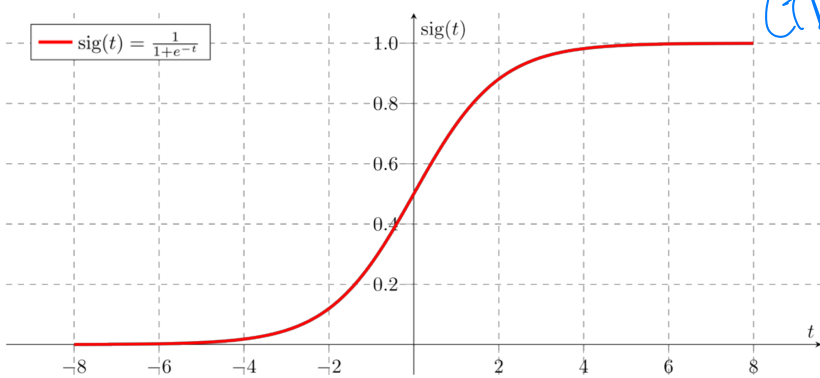
Linear Classifier

- **Logistic function (or Sigmoid function)**

$$y = \frac{1}{1+e^{-t}}$$

$$\boxed{\frac{dy}{dt} = y(1-y)}$$

GDM이용할거야!



$$\frac{dy}{dt} = \left(\frac{1}{1+e^{-t}}\right)^2 e^{-t}$$

$$= \left(\frac{1}{1+e^{-t}}\right)\left(\frac{e^{-t}}{1+e^{-t}}\right)$$

$$= \left(\frac{1}{1+e^{-t}}\right)\left(\frac{1+e^{-t}-1}{1+e^{-t}}\right)$$

$$= \left(\frac{1}{1+e^{-t}}\right)\left(1-\frac{1}{1+e^{-t}}\right)$$

$$= y(1-y)$$

Logistic Regression

- **Logistic Regression for a given boundary, $f(x_1,x_2)$**

Training Data		Model Output
(3, 1, 0)		$L(3,1) \approx 0$
(4, 3, 1)		$L(4,3) \approx 1$
(6, 1, 0)		$L(6,1) \approx 0$
(2, 3, 0)		$L(2,3) \approx 0$
(5, 10, 1)	→	$L(5,10) \approx 1$
(4, 8, 1)		$L(4,8) \approx 1$
(1, 2, 1)		$L(1,2) \approx 1$
(4, 4, 0)		$L(4,4) \approx 0$
(4, 1, 0)		$L(4,1) \approx 0$
(5, 5, 1)		$L(5,5) \approx 1$
…		

We determine the classes

$$L(x_1, x_2) = \frac{1}{1 + e^{-f(x_1, x_2)}}$$

when a linear boundary is given

$$f(x_1, x_2) = w_0 + w_1 x_1 + w_2 x_2$$

① $E = \sum_{(x,t)\in Data} (L(x) - t)^2$

Training of Logistic Regression

- **Then, How to find f(x_1,x_2)** → ① Define E ② Find w that minimize E

input (a, b, t) target v

$L(a,b) \approx t \Leftrightarrow (L(a,b) - t)^2$ is minimized

동치

Training Data

target value

(3, 1, 0)
(4, 3, 1)
(6, 1, 0)
(2, 3, 0)
(5, 10, 1)
(4, 8, 1)
(1, 2, 1)
(4, 4, 0)
(4, 1, 0)
(5, 5, 1)
…

We can determine the classes

$$L = \frac{1}{1 + e^{-f(x_1,x_2)}}$$

어떻게 찾을까?

How to obtain the boundary

$$f(x_1, x_2) = w_0 + w_1 x_1 + w_2 x_2$$

Model Output

close to의 의미

$L(3,1) \approx 0$
$L(4,3) \approx 1$
$L(6,1) \approx 0$
$L(2,3) \approx 0$
$L(5,10) \approx 1$
$L(4,8) \approx 1$
$L(1,2) \approx 1$
$L(4,4) \approx 0$
$L(4,1) \approx 0$
$L(5,5) \approx 1$

Training of Logistic Regression

- **Then, How to find f(x_1,x_2)**

가정

Training Data

(1, 3, 1, 0)
(1, 4, 3, 1)
(1, 6, 1, 0)
(1, 2, 3, 0)
(1, 5, 10, 1)
(1, 4, 8, 1)
(1, 1, 2, 1)
(1, 4, 4, 0)
(1, 4, 1, 0)
(1, 5, 5, 1)
…

We can determine the classes

$$L = \frac{1}{1 + e^{-\mathbf{wx}}}$$

How to obtain the boundary

$$f(x_0, x_1, x_2) = w_0x_0 + w_1x_1 + w_2x_2$$

$$f(x_0, x_1, x_2) = \mathbf{wx}$$

inner product

$$\mathbf{w} = (x_0, x_1, x_2), \mathbf{x} = (x_0, x_1, x_2)$$

Model Output

$L(1,3,1) \approx 0$
$L(1,4,3) \approx 1$
$L(1,6,1) \approx 0$
$L(1,2,3) \approx 0$
$L(1,5,10) \approx 1$
$L(1,4,8) \approx 1$
$L(1,1,2) \approx 1$
$L(1,4,4) \approx 0$
$L(1,4,1) \approx 0$
$L(1,5,5) \approx 1$

Training of Logistic Regression

MSE가 열열 경우 작동하지 않을수도

- **Let's do as we did for linear regression**
  - Define an error function, E
  - Find w that minimizes E by GDM

MSE : Mean (sum) Square Error

Hmm, it looks OK, but …

(1, 3, 1, 0)	$L(1,3,1) \approx 0$
(1, 4, 3, 1)	$L(1,4,3) \approx 1$
(1, 6, 1, 0)	$L(1,6,1) \approx 0$
(1, 2, 3, 0)	$L(1,2,3) \approx 0$
(1, 5, 10, 1)	$L(1,5,10) \approx 1$
(1, 4, 8, 1)	$L(1,4,8) \approx 1$
(1, 1, 2, 1)	$L(1,1,2) \approx 1$
(1, 4, 4, 0)	$L(1,4,4) \approx 0$
(1, 4, 1, 0)	$L(1,4,1) \approx 0$
(1, 5, 5, 1)	$L(1,5,5) \approx 1$
…	

E'

Mean (Sum)

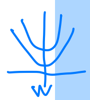
$$Error = \frac{1}{n} \sum_{(\mathbf{x},t)\in Data} (t - L(\mathbf{x};\mathbf{w}))^2$$

square

Error

왜 생략되어도 되나?

w값엔 영향이 없기에



where $L(\mathbf{x};\mathbf{w}) = \dfrac{1}{1+e^{-\mathbf{wx}}}$

not-linear of w

If E is minimized,
L will output like this!

Linear Classfier ; Linear boundary
" "
에 의해 데이터 주어짐

Training of Logistic Regression

- **Let's do as we did for linear regression**
  - We prefer to use $\boxed{\text{Cross Entropy}}$ for Logistic Regression
  - Find w that minimizes E

크로스 엔트로피 (수업에서 배우는건 Binary CrossEntropy)

(1, 3, 1, 0)
(1, 4, 3, 1)
(1, 6, 1, 0)
(1, 2, 3, 0)
(1, 5, 10, 1)
(1, 4, 8, 1)
(1, 1, 2, 1)
(1, 4, 4, 0)
(1, 4, 1, 0)
(1, 5, 5, 1)
...

그냥 암기..

$$E = \sum_{(x,t)\,\in \mathrm{Data}} -\left[t \log L(\mathrm{x};\mathrm{w}) + (1-t)\log(1-L(\mathrm{x};\mathrm{w}))\right]$$

where $L(\mathbf{x};\mathbf{w}) = \dfrac{1}{1+e^{-\mathbf{wx}}}$

How?

By Gradient Descent Method

Training of Logistic Regression

- **Formula** to find w

Find $\mathbf{w} = (w_0, w_1, \cdots, w_d)$ which minimizes

$$E(\mathbf{w}) = \sum_{(\mathbf{x},t) \in Data} - \left[t \log L(\mathbf{x}; \mathrm{w}) + (1-t) \log(1 - L(\mathbf{x}; \mathrm{w})) \right]$$

where

$$L(\mathbf{x}, \mathbf{w}) = \frac{1}{1 + e^{\vec{w} \cdot \vec{x}}}, \text{ and } \qquad f(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{d} \mathbf{w}_j \mathbf{x}_j$$

$Data$ is a set of $(\mathbf{x}, t)$ where $\mathbf{x} = (1, x_1, x_2, \cdots, x_d)$ and $t \in \{0,1\}$

$$f(\mathbf{x}, \mathbf{w}) = w_0 \cdot 1 + w_1 x_1 + w_2 x_2 + \cdots + w_d x_d$$

$$= \vec{w} \cdot \vec{x}$$

- How?
  - Using gradient descent method

Training of Logistic Regression

- **How to find a linear classifier**

$w_j^{t+1} = w_j^t - \eta \frac{\partial E}{\partial w_j}$

$$\frac{\partial}{\partial w_j} E(\mathbf{w}) = - \frac{\partial E}{\partial w_j} \sum_{(\mathbf{x},t) \in Data} - \left[t \log L(\mathbf{x}; \mathrm{w}) + (1-t) \log(1 - L(\mathbf{x}; \mathrm{w})) \right]$$

$$\text{where} \quad L(\mathbf{x}, \mathbf{w}) = \frac{1}{1 + e^{-f(\mathbf{x},\mathbf{w})}}, \quad f(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{d} w_j x_j$$

$$\frac{\partial E}{\partial w_j} = - \sum_{(\mathbf{x},t) \in Data} - \left[\frac{\partial}{\partial w_j} \left(t \log L(\mathbf{x}; \mathbf{w}) + (1-t) \log(1 - L(\mathbf{x}; \mathbf{w})) \right) \right]$$

$$\frac{\partial E}{\partial w_j} = - \sum_{(\mathbf{x},t) \in Data} - \left[\frac{\partial}{\partial w_j} \left(t \log L(\mathbf{x}; \mathbf{w}) \right) + \frac{\partial}{\partial w_j} \left((1-t) \log(1 - L(\mathbf{x}; \mathbf{w})) \right) \right]$$

Training of Logistic Regression

- **How to find a linear classifier**

$$\frac{\partial E}{\partial w_j} = \sum_{(\mathbf{x},t)\in Data} -\left[\frac{\partial}{\partial w_j}\left(t\log L(\mathbf{x};\mathbf{w}) + (1-t)\log(1-L(\mathbf{x};\mathbf{w}))\right)\right]$$

$$\frac{\partial E}{\partial w_j} = \sum_{(\mathbf{x},t)\in Data} -\left[\boxed{\frac{\partial}{\partial w_j}(t\log L(\mathbf{x};\mathbf{w}))} + \frac{\partial}{\partial w_j}\left((1-t)\log(1-L(\mathbf{x};\mathbf{w}))\right)\right]$$

$$t\log L(\mathbf{x};\mathbf{w}) = t\log\frac{1}{1+e^{-f(\mathbf{x};\mathbf{w})}}$$

$$g(h) = t\log h$$

$$h(f) = \frac{1}{1+e^{-f}} \rightarrow \boxed{\frac{\partial h}{\partial f} = h(1-h)}$$

$$f(\mathbf{x},\mathbf{w}) = w_0\cdot 1 + w_1x_1 + \cdots + w_dx_d$$

합성함수

$$g(h(f)) = t\log\frac{1}{1+e^{-f}} = t\log L(\mathbf{x};w)$$

$$\frac{\partial g}{\partial w_j} = \frac{\partial g}{\partial h}\frac{\partial h}{\partial f}\frac{\partial f}{\partial w_j} = \frac{t}{\cancel{h}}\cdot\cancel{h}\cdot(1-h)\cdot x_j = t\cdot(1-h)\cdot x_j$$

Training of Logistic Regression

- **How to find a linear classifier**

$$\frac{\partial E}{\partial w_j} = \sum_{(\mathbf{x},t)\in Data} -\left[\frac{\partial}{\partial w_j}\left(t \log L(\mathbf{x};\mathbf{w}) + (1-t)\log(1-L(\mathbf{x};\mathbf{w}))\right)\right]$$

$$\frac{\partial E}{\partial w_j} = \sum_{(\mathbf{x},t)\in Data} -\left[\frac{\partial}{\partial w_j}\left(t \log L(\mathbf{x};\mathbf{w})\right) + \frac{\partial}{\partial w_j}\left((1-t)\log(1-L(\mathbf{x};\mathbf{w}))\right)\right]$$

$$(1-t)\log\left(1-L(\mathbf{x};\mathbf{w})\right) = (1-t)\log\left(1-\frac{1}{1+e^{-f(\mathbf{x};\mathbf{w})}}\right)$$

$$g(h) = (1-t)\log(1-h)$$

$$h(f) = \frac{1}{1+e^{-f}}$$

$$f(\mathbf{x},\mathbf{w}) = w_0 \cdot 1 + w_1 x_1 + \cdots + w_d x_d$$

$$g(h(f)) = (1-t)\log\left(1-\frac{1}{1+e^{-f}}\right)$$

$$= (1-t)\log\left(1-L(\mathbf{x};\mathbf{w})\right)$$

$$\frac{\partial g}{\partial w_j} = \frac{\partial g}{\partial h}\frac{\partial h}{\partial f}\frac{\partial f}{\partial w_j} = -\frac{1-t}{1-h}\cdot h\cdot(1-h)\cdot x_j = (t-1)\cdot h\cdot x_j$$

Training of Logistic Regression

- **How to find a linear classifier**

$f(\mathbf{x};\mathbf{w})$

$$h(f) = \frac{1}{1+e^{-f}}$$

$$\frac{\partial E}{\partial w_j} = \sum_{(\mathbf{x},t)\in Data} -\left[\frac{\partial}{\partial w_j}\left(t\log L(\mathbf{x};\mathrm{w}) + (1-t)\log(1-L(\mathbf{x};\mathrm{w}))\right)\right]$$

$$\frac{\partial E}{\partial w_j} = \sum_{(\mathbf{x},t)\in Data} -\left[\boxed{\frac{\partial}{\partial w_j}\left(t\log L(\mathbf{x};w)\right)} + \boxed{\frac{\partial}{\partial w_j}\left((1-t)\log(1-L(\mathbf{x};\mathrm{w}))\right)}\right]$$

$$\frac{\partial E}{\partial w_j} = \sum_{(\mathbf{x},t)\in Data} -\left[t\cdot(1-h)\cdot x_j + (t-1)\cdot h\cdot x_j\right] = \sum_{(\mathbf{x},t)\in Data} \boxed{(h-t)\cdot x_j}$$

각데이터에대한 기울기 축적

$$\frac{\partial E}{\partial w_j} = -\sum_{(\mathbf{x},t)\in Data}\left(t - L(\mathbf{x};\mathbf{w})\right)x_j$$

$$L(\mathbf{x},\mathbf{w}) = \frac{1}{1+e^{-f(\mathbf{x},\mathbf{w})}}, \quad f(\mathbf{x},\mathbf{w}) = \sum_{j=0}^{d} w_j x_j$$

Training of Logistic Regression

- **GDM Algorithm**

Randomly choose an initial solution, $\mathbf{w}^0 = \left(w_0^0, w_1^0, \cdots, w_d^0\right)$

$t = 0$

Repeat

for j = 0, …, d // for all w_j

$g_j^t = 0$

for i = 1, …, n // for all data, $\mathbf{x}_i = (x_{i1}, x_{i2}, \cdots, x_{id})$

$$g_j^t = g_j^t - \left(t_i - L(\mathbf{x}_i; \mathbf{w}^t)\right)x_{ij}$$

$w_j^{t+1} = w_j^t - \eta g_j^t$

Gradient accumulate

기울기 축적

$t = t + 1$

Until stopping condition is satisfied

Logistic Regression

Some Questions

- **Why not MSE?**

MSE will be O.K!

except some special cases

- **What is Cross Entropy?**

$-(t\log h+(1-t)\log(1-h))$

what? outside에 boundary가 생기는 경우

why? GDM이 작동하지 X

Some Questions

- **Why not MSE?**
  - Find w to minimize E

non-linear of w

Hmm, it looks GOOD, why not?

(1, 3, 1, 0)
(1, 4, 3, 1)
(1, 6, 1, 0)
(1, 2, 3, 0)
(1, 5, 10, 1)
(1, 4, 8, 1)
(1, 1, 2, 1)
(1, 4, 4, 0)
(1, 4, 1, 0)
(1, 5, 5, 1)
…

$$Error = \frac{1}{2} \sum_{(\mathbf{x},t) Data} (t - L(\mathbf{x};\mathbf{w}))^2$$

지수항으로 들어가있음

where $L(\mathbf{x};\mathbf{w}) = \boxed{\dfrac{1}{1 + e^{-\mathbf{wx}}}}$

$L(1,3,1) \approx 0$
$L(1,4,3) \approx 1$
$L(1,6,1) \approx 0$
$L(1,2,3) \approx 0$
$L(1,5,10) \approx 1$
$L(1,4,8) \approx 1$
$L(1,1,2) \approx 1$
$L(1,4,4) \approx 0$
$L(1,4,1) \approx 0$
$L(1,5,5) \approx 1$

Since L is not linear wrt w,
we have to use GDM

L(a,b)≃0

(a,b) L(c,d)≃0

(직선이 바깥쪽에 있다면)

(c,d)

Some Questions

- **Why not MSE?**

h(L(f)) 이런구조

$$\frac{\partial}{\partial w_j}E(\mathbf{w}) = \boxed{\frac{1}{2}}\sum_{i=1}^{n}\frac{\partial}{\partial w_j}\underbrace{\left(t - L(\mathbf{x}_i,\mathbf{w})\right)^2}_{h}$$

$$= \frac{1}{2}\sum_{i=1}^{n}\frac{\partial f(\mathbf{x}_i,\mathbf{w})}{\partial w_j}\frac{\partial L(\mathbf{x}_i,\mathbf{w})}{\partial f(\mathbf{x}_i,\mathbf{w})}\frac{\partial\left(t - L(\mathbf{x}_i,\mathbf{w})\right)^2}{\partial L(\mathbf{x}_i,\mathbf{w})}$$

$$= \sum_{i=1}^{n}\mathbf{x}_i\underbrace{L(\mathbf{x}_i,\mathbf{w})\left(1 - L(\mathbf{x}_i,\mathbf{w})\right)}_{y(1-y)}\left(t - L(\mathbf{x}_i,\mathbf{w})\right)$$

$$f(\mathbf{x}_i,\mathbf{w}) = \mathrm{w}\mathbf{x}_i$$

$$\frac{\partial f(\mathbf{x}_i,\mathbf{w})}{\partial w_j} = x_{ij}$$

$$\frac{\partial L(\mathbf{x}_i,\mathbf{w})}{\partial f(\mathbf{x}_i,\mathbf{w})} = L(\mathbf{x}_i,\mathbf{w})\left(1 - L(\mathbf{x}_i,\mathbf{w})\right)$$

$$\frac{\partial\left(t - L(\mathbf{x}_i,\mathbf{w})\right)^2}{\partial L(\mathbf{x}_i,\mathbf{w})} = \mathbf{2}\left(t - L(\mathbf{x}_i,\mathbf{w})\right)$$

$$w_j^{t+1} = w_j^t - \eta\sum_{i=1}^{n}x_{ij}\underbrace{L(\mathbf{x}_i,\mathbf{w}^t)}_{1}\underbrace{\left(1 - L(\mathbf{x}_i,\mathbf{w}^t)\right)}_{0}\underbrace{\left(t - L(\mathbf{x}_i,\mathbf{w}^t)\right)}_{\text{something}}$$

1 × 0 × something ⇒ 0

GDM의 가정

w에 따라 달라지기에 매우 복잡..

- If w is randomly initialized but wrongly initialized, causing L to output 0 or 1 for all the training samples?
- What will happen?

boundary가 outside에 있는걸

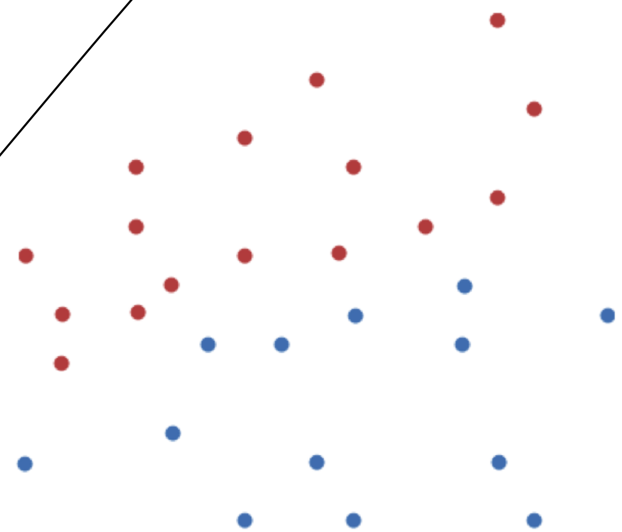
기울기 축적값이 항상0 → w가 갱신되지X

이런경우 ⇒ L(모든x)=0/1

Some Questions

- **Why not MSE?**
  - Wrongly initialized?

→ MSE에서 작동 X



$$L(x_1, x_2) = \frac{1}{1 + e^{-f(x_1, x_2)}}$$

Some Questions

- **What is Cross Entropy?**
  - We can regard the output of L as Probability

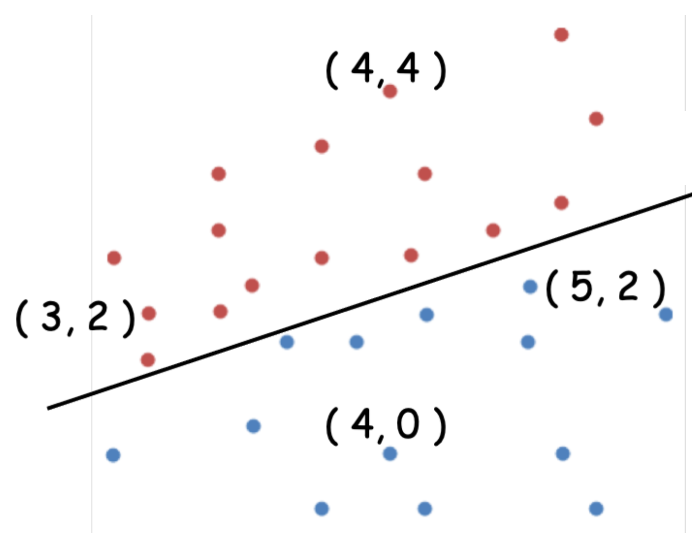
$$L(x_1, x_2) = \frac{1}{1 + \exp^{-f(x_1,x_2)}} \longleftarrow$$

estimate

We may "regard" it as the probability to be RED

$f(x_1, x_2) = -x_1 + x_2 + 2$

	Label	t	$f(x_1, x_2)$	$L(x_1, x_2)$
(4, 10)	Red	1	8	0.9997
(4, 4)	Red	1	2	0.8808
(3, 2)	Red	1	1	0.7311
(4, 2)	?	?	0	0.5000
(4, 0)	Blue	0	-2	0.1192
(5, 2)	Blue	0	-3	0.0474
(4,-6)	Blue	0	-8	0.0003



Some Questions

- **What is Cross Entropy?**
  - The problem to optimize for the model training

정확하게 예측

Find w so that the model correctly predicts all training data

최대화 확률

⟺ Find w which maximizes the probability that the model correctly predicts all training data

⟺ Find w which maximizes the following:

argmax w

Red Blue 곱셈

$$\left(\prod_{(x,1)\in Data} L(x;w)\right) \times \left(\prod_{(x,0)\in Data} (1 - L(x;w))\right)$$

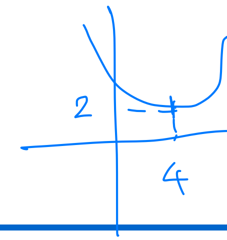
Red Red에 대한 확률 Blue Blue에 대한 확률

Prob. that L says Red when Red is given

Prob. that L says Blue when Blue is given

To train L.R

min f(x)=2
argmin f(x)=4
↓
find x



Some Questions

- **What is Cross Entropy?**

$$\underset{w}{\operatorname{argmax}}\left(\prod_{x \text{ is RED}} L(x;w) \times \prod_{x \text{ is BLUE}} (1 - L(x;w))\right)$$

← We want to find w to maximize the probability that the model correctly predicts all training data

확률

곱셈

$$= \underset{w}{\operatorname{argmax}} \log\left(\prod_{x \text{ is RED}} L(x;w) \times \prod_{x \text{ is BLUE}} (1 - L(x;w))\right)$$

$\underset{w}{\operatorname{argmax}}\ f(w) = \underset{w}{\operatorname{argmax}} \log f(w)$
$(f(w)>0)$

로그취함

∵ log는 증가함수라서 max되는지점의 w값이 f(w)의 max지점의 w값과 같다

(f(w)가 최대면 log f(w)도 최대)

$$= \underset{w}{\operatorname{argmax}}\left(\sum_{x \text{ is RED}} \log L(x;w) + \sum_{x \text{ is BLUE}} \log(1 - L(x;w))\right)$$

덧셈

$$= \underset{w}{\operatorname{argmax}}\left(\sum_{(x,t=1)\in \mathrm{Data}} \log L(x;w) + \sum_{(x,t=0)\in \mathrm{Data}} \log(1 - L(x;w))\right)$$

trick (중요하지 않음)

$$= \underset{w}{\operatorname{argmax}}\left(\sum_{(x,t=1)\in \mathrm{Data}} t\log L(x;w) + (1-t)\log(1 - L(x;w)) + \sum_{(x,t=0)\in \mathrm{Data}} t\log L(x;w) + (1-t)\log(1 - L(x;w))\right)$$

대입시 0이됨   대입시 0이됨

$$= \underset{w}{\operatorname{argmax}}\left(\sum_{(x,t)\in \mathrm{Data}} t\log L(x;w) + (1-t)\log(1 - L(x;w))\right)$$

merge!

$$= \underset{w}{\operatorname{argmin}}\left(\sum_{(x,t)\in \mathrm{Data}} \boxed{-[t\log L(x;w) + (1-t)\log(1 - L(x;w))]}\right)$$

→ cross entropy

이공식을 최소화 = 확률의최대화

Some Questions

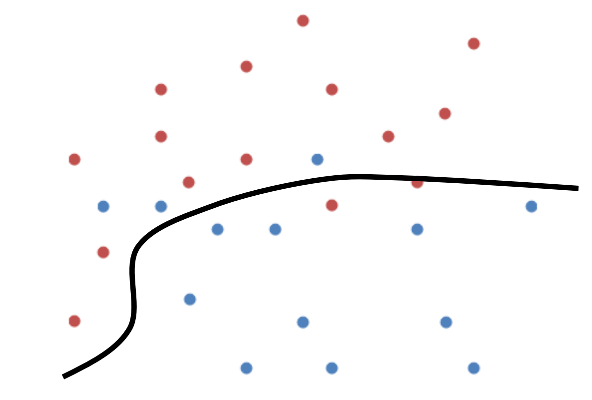
- **MSE**
  - Measures the difference between model outputs and target values.
  - Suitable when the outputs are continuous real numbers.
  - Training: to find *w* that minimize the error.
- **Cross Entropy**
  - Measures the likelihood (probability) that the model's predictions are correct.
  - Suitable when the outputs can be interpreted as probabilities.
  - Training: to find *w* that minimize cross entropy
    - Equivalent to maximize the probability of correct predictions.

동일하다

Discussion

- **Logistic Regression**
  - Optimized by Gradient Descent method
  - A linear boundary
    - What about non-linear boundary?

GDM으로 다구할수있다.



$$f(x_1, x_2) = w_0 + w_1 x_1^2 + w_2 x_2^2 + w_3 x_1 x_2 + w_4 x_1 + w_5 x_2$$

  - Binary classifier
    - If we have more than 3 classes…