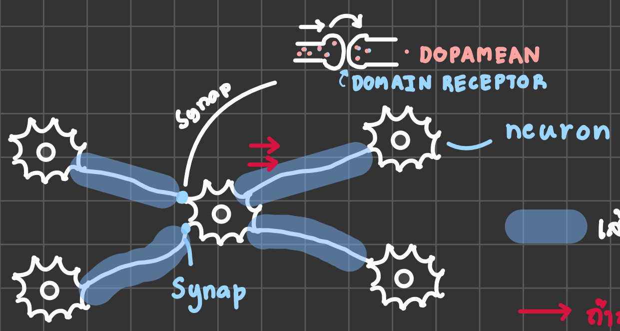
Artificial Intelligence
A Modern Approach

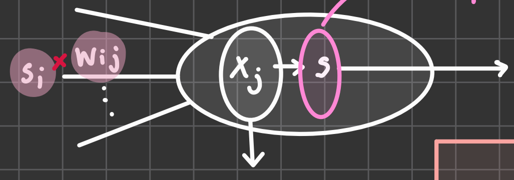
KAMPANART

Artificial Neurons



เส้นแต่ละเส้นมีความสำคัญต่างกัน !

→ ถ้าการรวมกันของ signal มากพอจะส่ง activate (fire !)



เป็นตัว perform ว่า signal มากพอไหม ? (> 0)

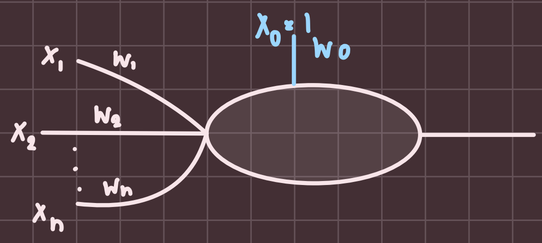
→ s_j (สัญญาณขาออกของ node j ใดๆ)

ตัวรับข้อมูลเข้า

$$x_j = \boxed{\sum_{i=1}^{n} w_{ij} s_i} + \boxed{\theta_j}$$

w_{ij} (Weight i & j) : weight ที่เชื่อมโยงระหว่าง neuron i และ j

● = Input signal



$$w_1x_1 + w_2x_2 + \dots + w_nx_n > b$$

$$\underbrace{-b + w_1x_1 + w_2x_2 + \dots + w_nx_n} > 0$$

$$\boxed{-b} + \boxed{\sum_{i=1}^{n} w_i x_i} > 0$$

bias

สมมติ $w_0 = -b$: $w_0 x_0 + \sum_{i=1}^{n} w_i x_i > 0$

$(x_0 = 1)$

$$\downarrow$$

$$\sum_{i=0}^{n} w_i x_i > 0$$

Signal function : — binary , linear, sigmoidal , Gassion, Relu

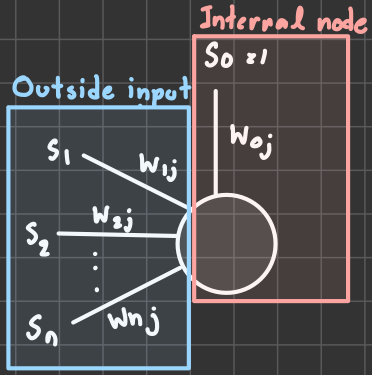
Neuron Equation

Binary Threshold Signal Function

$$S(x_j) = \begin{cases} 1 & x_j \geq 0 \\ 0 & x_j < 0 \end{cases} \qquad S(x_j) = \begin{cases} 1 & x_j \geq 0 \\ -1 & x_j < 0 \end{cases}$$

$x_j \downarrow \sum_{i=0} w_{ij} s_i$

binary | bipolar



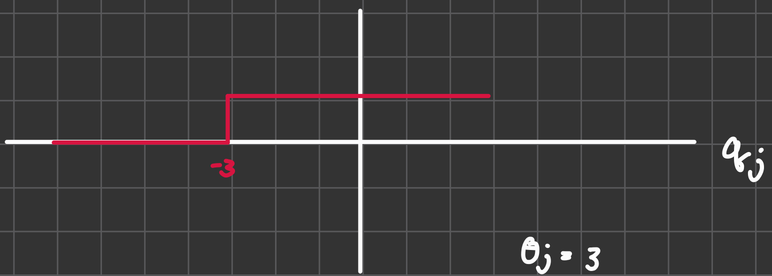
$$q_j = \sum_{i=1}^{n} w_{ij} s_i$$

$$x_j = q_j + \theta_j$$

$\downarrow$

+3 มาเติม

$$x_j = \underbrace{\sum_{i=1}^{n} w_{ij} s_i}_{\text{outside input}} + \underbrace{3}_{\text{internal node}} \quad (\theta_j = 3)$$



$\theta_j = 3$

เราใช้ trian NN เป็นรอบๆ (discrete time)

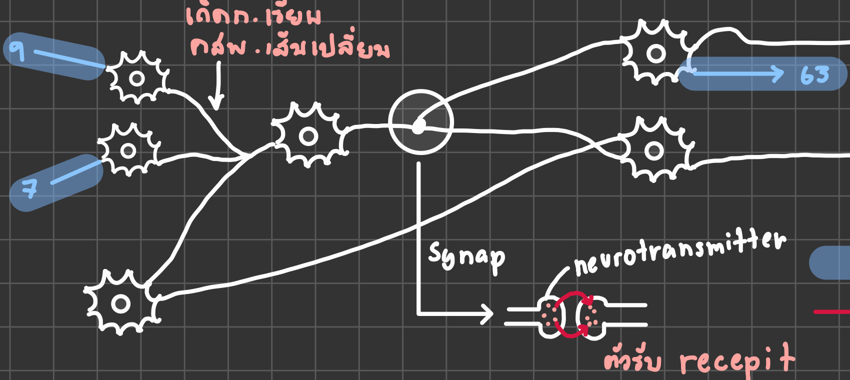
Threshold Logic Neuron (TLN) in Discrete time

$$S(x_j \text{ รอบ } k+1) = \begin{cases} 1 & x_j\ k+1 > 0 \\ S(x_j \text{ รอบ } k) & x_j\ k+1 = 0 \\ 0 & x_j\ k+1 < 0 \end{cases}$$

(เอา signal เดิมมาออก)

extended เป็น bipolar ได้

13 Jan 2024 . Artificial Neuron



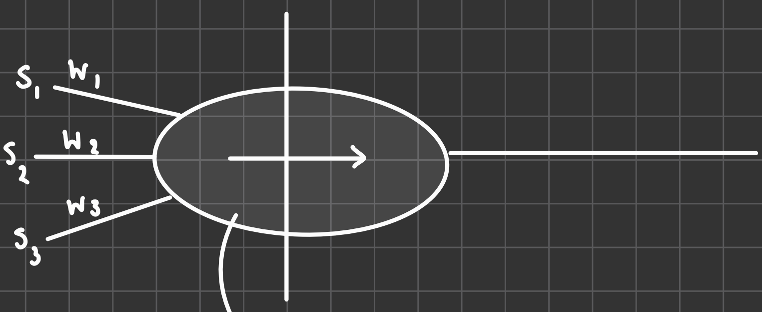
เส้นแต่ละเส้นมีความสำคัญต่างกัน !

→ ถ้าการรวมกันของ signal มากพอจะส่ง activate (fire!)

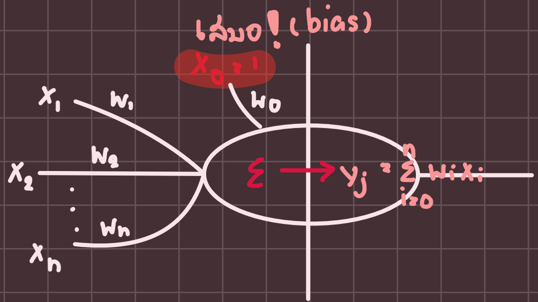
Elasticity , Plasticity

Elasticity: พึ่งเรียนแล้วลืม (กลับจุดเดิม)

Plasticity: เกิดการทำซ้ำ



$y_j = w_1 s_1 + w_2 s_2 + \dots + w_n s_n$



$w_1 x_1 + w_2 x_2 + \dots + w_n x_n > b$

$-b + w_1 x_1 + w_2 x_2 + \dots + w_n x_n > 0$

$w_0 x_0 + w_1 x_1 + w_2 x_2 + \dots + w_n x_n > 0$

$$S(n_j) = \begin{cases} 1 & \text{if } \sum_{i=1}^{n} w_i x_i > b \\ 0 & \text{otherwise} \end{cases}$$

$$\text{Output } S(y_j) = \begin{cases} 1 & \text{if } \sum_{i=0}^{n} n_i x_i \boxed{\geq} 0 \\ 0 & \text{otherwise} \end{cases}$$

? >, ≥ depends on user

Signal Function

เราใช้ trian NN เป็นรอบ ๆ (discrete time)

Threshold Logic Neuron (TLN) in Discrete time

$$S(x_j \text{ รอบ } k+1) = \begin{cases} 1 & x_j\ k+1 > 0 \\ S(x_j \text{ รอบ } k) & x_j\ k+1 = 0 \\ 0 & x_j\ k+1 < 0 \end{cases}$$

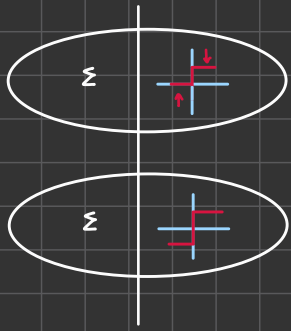
(เอา signal เดิมมาออก)

extended เป็น bipolar ได้

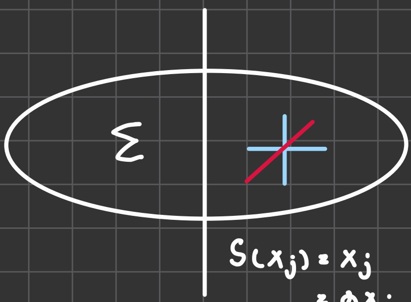
TLN perceptron (neuron) → {0,1} binary

$\begin{cases} 1 \\ -1 \end{cases}$

→ {1,-1} bipolar



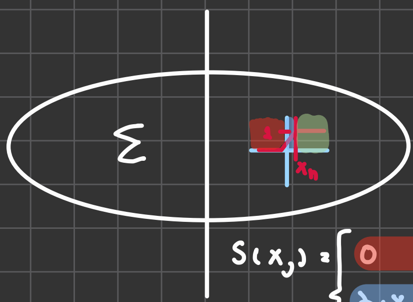
Linear Signal



$S(x_j) = x_j$
$= a x_j$

shift x-axis
$a(x-c)+b$
shift y-axis
$ax+b$

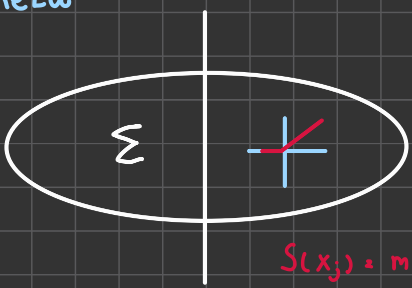
Linear Threshold Signal



$$S(x_j) = \begin{cases} 0 & x_j \le 0 \\ a_j x_j & 0 < x_j < x_m \\ 1 & x_j \ge x_m \end{cases}$$

$S(x_j) = \max(0, \min(a_j x_j, 1))$

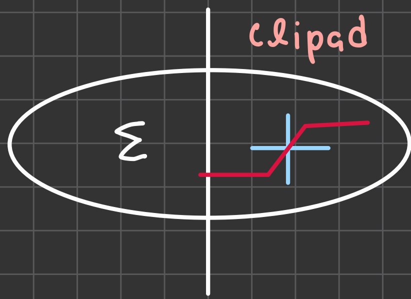
ReLu



shift x-axis
$\max(0-c)$
shift y-axis
$\max(0-c)+b$

$$S(x_j) = \max(0, x_j) = \begin{cases} 0 & \text{if } x \le 0 \\ x & \text{if } x > 0 \end{cases}$$

Clipad

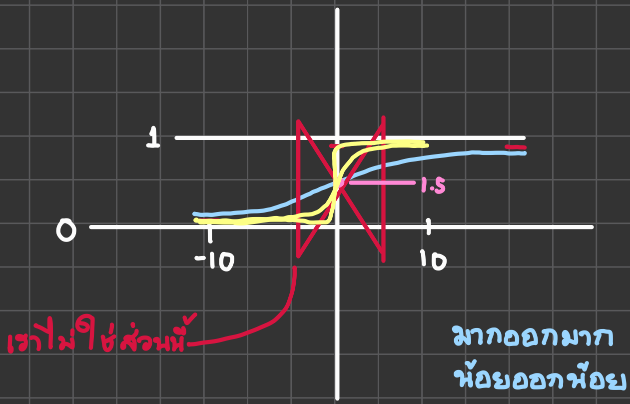


Sigmoid [Perceptron]

$$\Sigma(y_j) = \frac{1}{1+e^{-y_j}} = \frac{1}{1+e^{-\lambda_j y_j}}$$

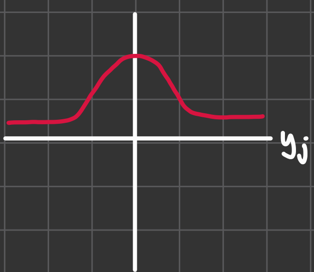
- ยิ่ง λ มีค่ามากกราฟก็จะชัน (เส้นเหลือง)
ยิ่งชันมากยิ่งเข้าใกล้พฤติกรรม binary

∴ monotonic
continuous
bounded

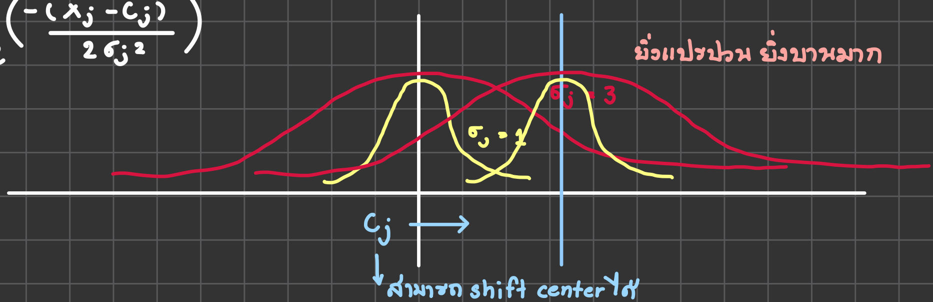


Gousian Signal Function

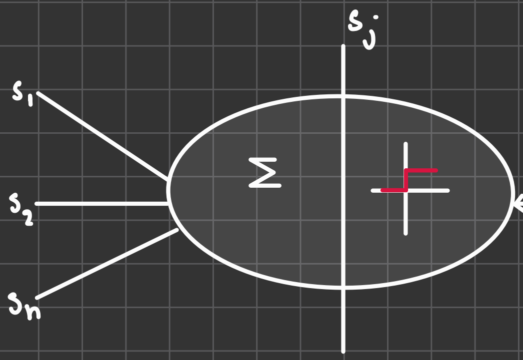
(non-monotonic) อยากได้ output ที่ใกล้ center มากสุด ใกล้center ออก max, 1 (ตรงกับค.ต้องการเท่านั้นออก max)



$$S(x_j) = e^{\left(\frac{-(x_j - c_j)^2}{2\sigma_j^2}\right)}$$



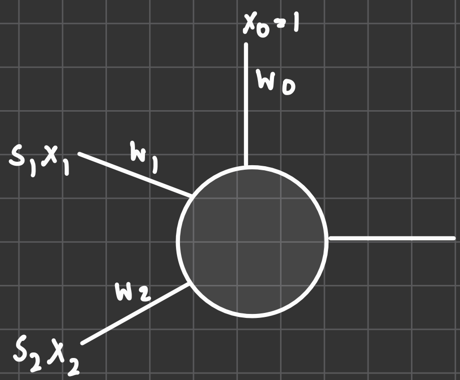
20 Jan 2025: TLN Architecture



output = $\begin{cases} 1 & \text{if } y_j > 0 \\ 0 & \text{otherwise} \end{cases}$ classification

output ∈ {0,1} → binary

$y_j = \sum_{i=0} w_i s_i$



$w_1 x_1 + w_2 x_2 > b$ (มากกว่า treshow ค่าหนึ่งจึงจะ activate)

$-b + w_1 x_1 + w_2 x_2 > 0$

$w_0 + w_1 x_1 + w_2 x_2 > 0$

เราเริ่มเอา neural ก้อนนี้ไป classify logic And

AND TABLE : +, T = 1, | −, F = 0

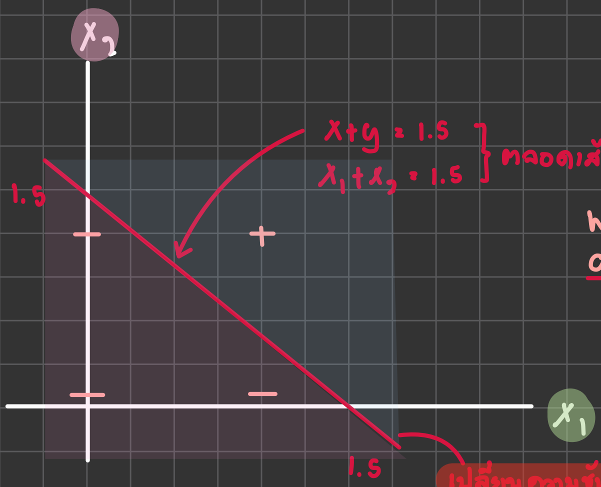
	x_1	x_2	target (output)		
x_1	0	0	−	→ $w_0 + w_1(0) + w_2(0) < 0$	$w_0 < 0$
x_2	0	1	−	→ $w_0 + w_1(0) + w_2(1) < 0$	$w_0 + w_2 < 0$
x_3	1	0	−	→ $w_0 + w_1(1) + w_2(0) < 0$	$w_0 + w_1 < 0$
x_4	1	1	+	→ $w_0 + w_1(1) + w_2(1) > 0$	$w_0 + w_1 + w_2 > 0$

target = − (false)

target = + (true)

หา set ของ $w_0 w_1 w_2$: $\vec{w} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$ $\vec{x}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ $\vec{w} \cdot \vec{x}$

หน้าที่หลักคือการหาเส้นนี้



$x + y = 1.5$ } ตลอดเส้น
$x_1 + x_2 = 1.5$

$w_0 + w_1 x_1 + w_2 x_2 = 0$

$x_2 = -\frac{w_1}{w_2} x_1 - \frac{w_0}{w_2}$

$ax + by + c = 0$

$y = \frac{-a}{b} x - \frac{c}{b}$

ความชัน

จุดตัดแกน.

$w_0 + w_1 x + w_2 y > 0$

$a_x + by + c > 0$ (สมการเชิงเส้น)

ถ้าอยากรู้ว่า > 0 ตรงไหนก็หาเส้นตรง

เปลี่ยนความชันได้ (มี infinite เส้น); เส้นนี้ suppose มาเพื่อความง่าย.

จากสมการ $w_0 + w_1 x + w_2 y > 0$ จะได้ว่า

$x_1 + x_2 > 1.5$

$-1.5 + x_1 + x_2 > 0$

$-1.5 + (1)x_1 + (1)x_2 > 0$

$\vec{w} = \begin{bmatrix} -1.5 \\ 1 \\ 1 \end{bmatrix}$

$x_1 + x_2 < 1.5$

$1.5 - x_1 - x_2 > 0$

$1.5 + (-1)x_1 + (-1)x_2 > 0$

$\vec{w} = \begin{bmatrix} 1.5 \\ -1 \\ -1 \end{bmatrix}$

---

	x_1	x_2	target (output)
$\vec{x}_1$	0	0	−
$\vec{x}_2$	0	1	−
$\vec{x}_3$	1	0	−
$\vec{x}_4$	1	1	+

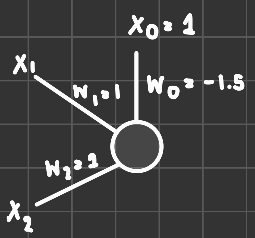
เลือกใช้สมการ $-1.5 + x_1 + x_2 > 0$

$-1.5 + 0 + 0 = -1.5 > 0$; FALSE (−)

$-1.5 + 0 + 1 = -0.5 > 0$; FALSE (−)

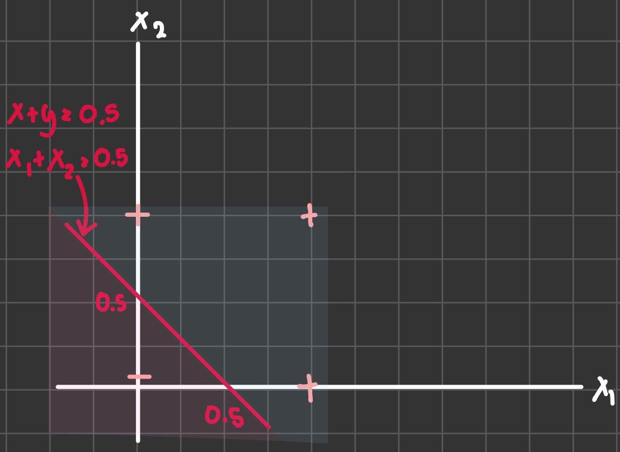
$-1.5 + 1 + 0 = -0.5 > 0$; FALSE (−)

$-1.5 + 1 + 1 = 0.5 > 0$; TRUE (+)



OR TABLE : +, T = 1 | −, F = 0

	x_1	x_2	target (output)
x^1	0	0	−
x^2	0	1	+
x^3	1	0	+
x^4	1	1	+



จะได้ $x_1+x_2 > 0.5$

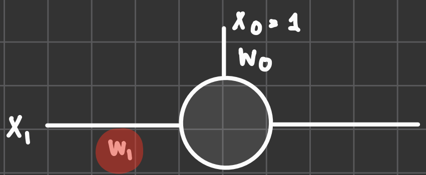
$-0.5+x_1+x_2 > 0$ ✓

$\vec{w} = \begin{bmatrix} -0.5 \\ 1 \\ 1 \end{bmatrix}$

$x_1+x_2 < 0.5$

$0.5 - x_1 - x_2 > 0$

$\vec{w} = \begin{bmatrix} 0.5 \\ -1 \\ -1 \end{bmatrix}$



เข้า 0 อยากให้มากกว่า 1

เข้า 1 อยากให้น้อยกว่า 0

x_1	target
0	+
1	−

จาก $w_0 + w_1x_1 > 0$ จะได้ว่า

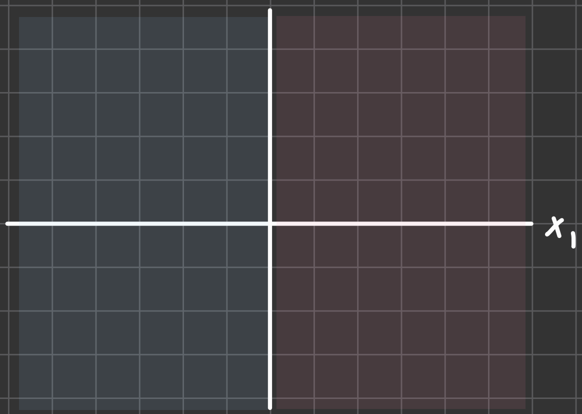
$w_0 + w_1(0) > 0$

$w_0 + w_1(1) \leq 0$

$\hookrightarrow < 0$

; $w_0 > 0$

$w_0 + w_1 \leq 0$



$x_1 = 0$

(เข้าลบออกบวก)

$x_1 < 0$

$w_0 + w_1x_1 < 0$

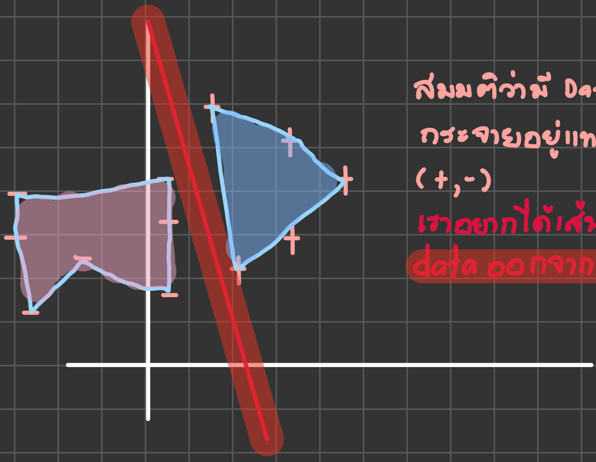
$-\cancel{w_0} - w_1x_1 > 0$

$x_1 > 0$

$w_0 + w_1x_1 > 0$

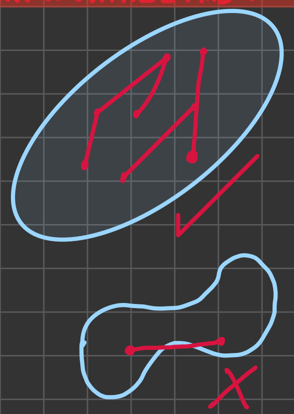
$\vec{w} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$

Linearly Seperable : สามารถจำแนกได้โดยเส้นตรง ลงไป 1 เส้น



สมมติว่ามี Data กระจายอยู่แทนด้วย (+,-)
เราอยากได้เส้น split data ออกจากกัน

สร้างพท.รอบกลุ่มตย. ที่เล็กที่สุด (intersection) ที่สามารถครอบ AREA ได้ Convex house

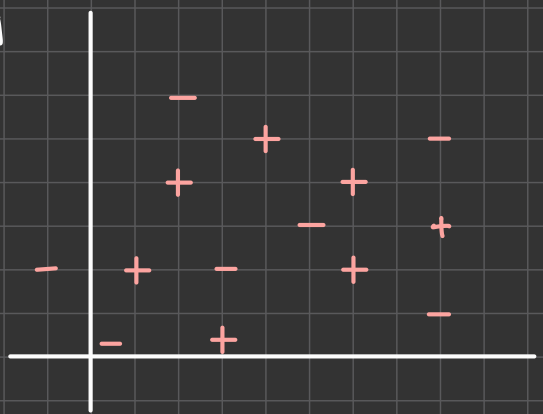


โดยที่เส้นที่เชื่อมต้อง ไม่ออกจากพท.
และ convex house ของทั้ง 2 classes จะต้องไม่ทับกัน (intersect แล้ว เป็น empty area)

Non-Linearly Separable

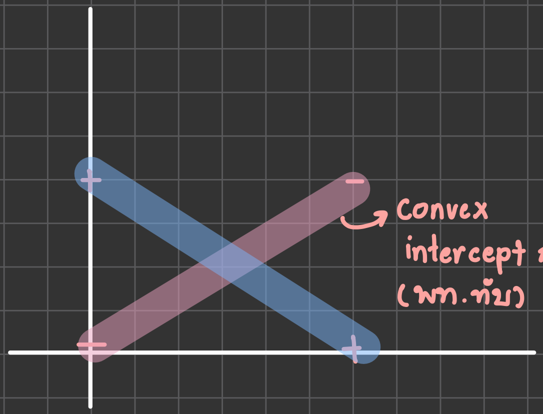
Linearly Non-Separable
(ยึดคำนี้)

TLN



มีจุดกระจายตัวใน space
จำนวน P จุด กระจายใน n มิติ
← 14 จุด กระจายใน 2 มิติ
อยากรู้ว่าโอกาสเป็น linearly separable = ?

แล้วถ้าแยกไม่ได้โดยเส้นตรง (Linearly Non-Separable)

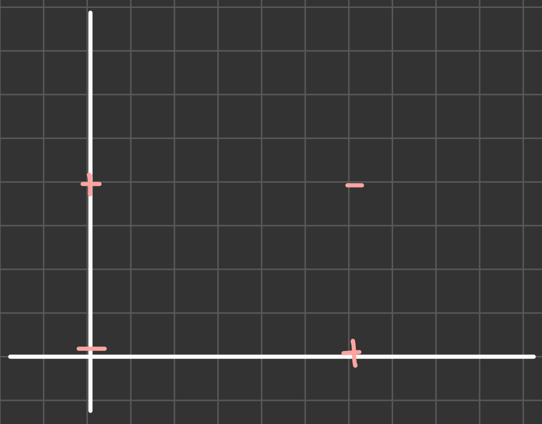


convex intercept กัน (พท.ทับ)

XOR TABLE (แยกไม่ได้โดย linearly)

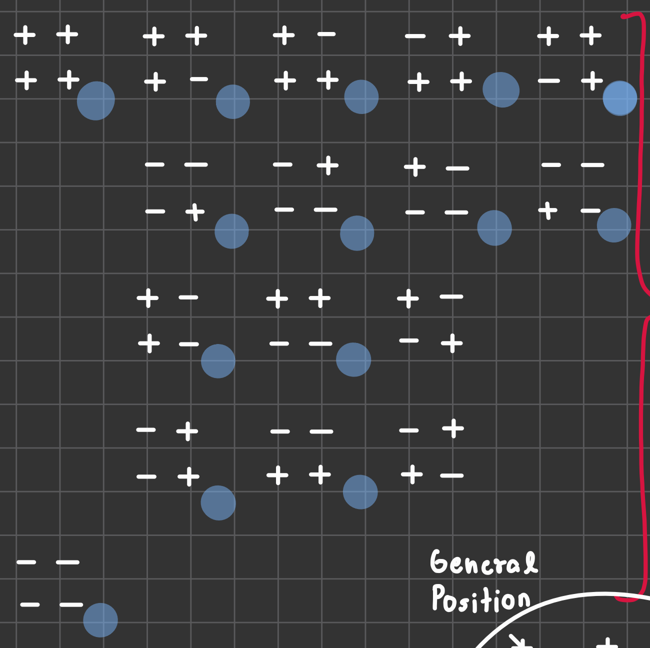
x_1	x_2	target
0	0	-
0	1	+
1	0	+
1	1	-

(เพศเดียวกันไม่สามารถมีลูกได้)



มีจุด 4 จุดใน space 2 มิติ

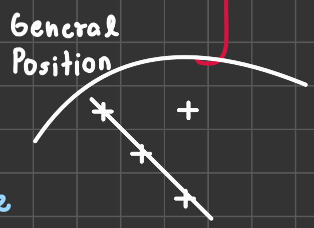
Assign ค่าได้ 16 แบบ



16 แบบ

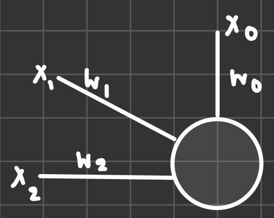
$P = \frac{14}{16}$

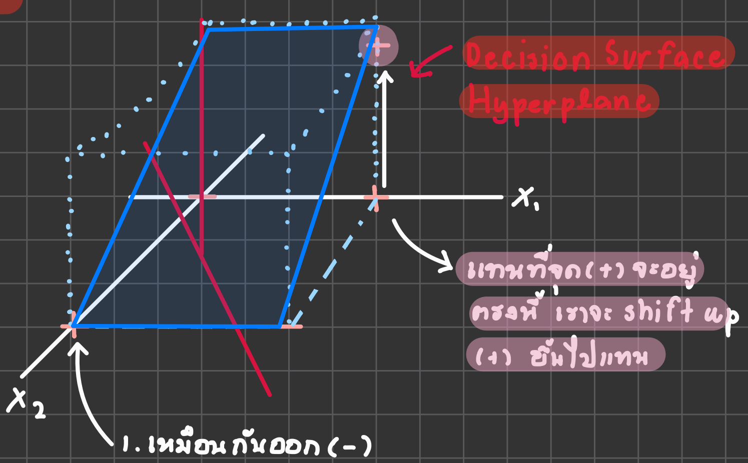
ค่าความจุของ perceptron!



● Linearly Seperable

ถ้าแยกไม่ได้ทำอย่างไร ?



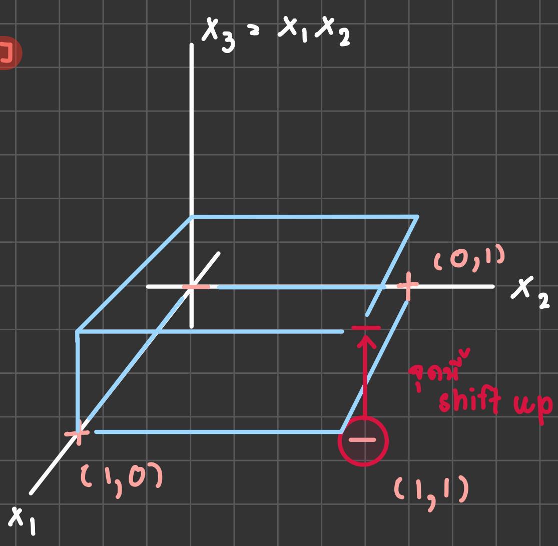


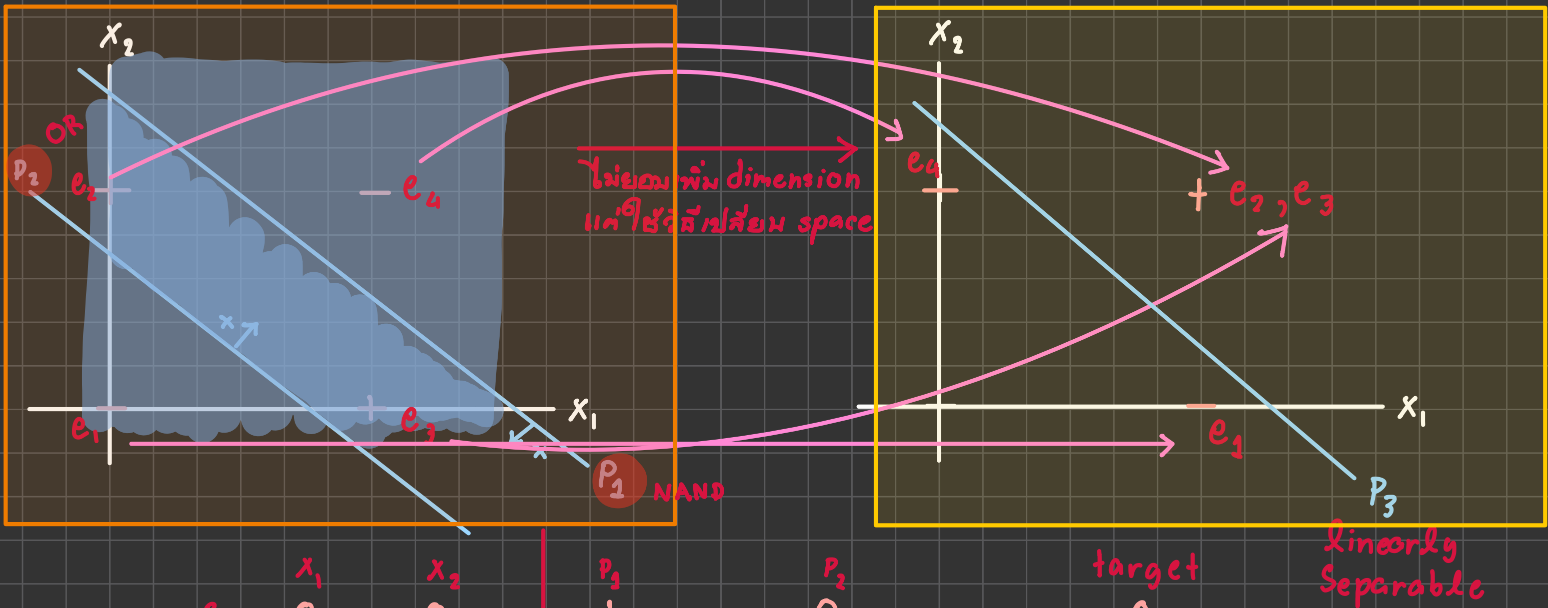
เพิ่ม Dimension โดยการจับ $x_1 \cdot x_2$ [+ = 1 , - = 0]

x_1	x_2	$x_1 \cdot x_2$ (2)	target	
0	0	0	0	−
0	1	0	1	+
1	0	0	1	+
1	1	1	0	−

$w_0 + w_1 x_1 + w_2 x_2 + w_3(x_1 \cdot x_2) > 0$

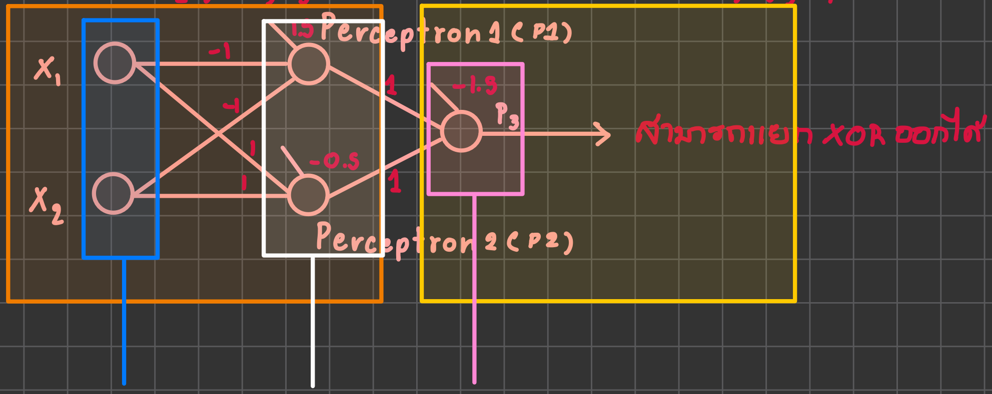
(ระนาบ decision surface)



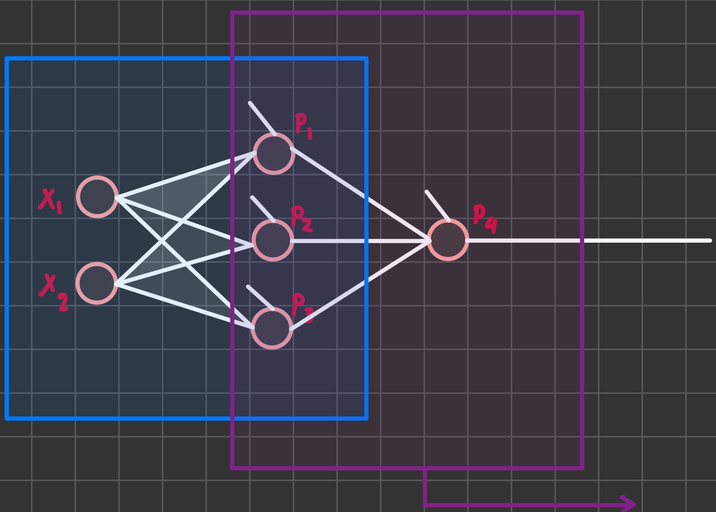


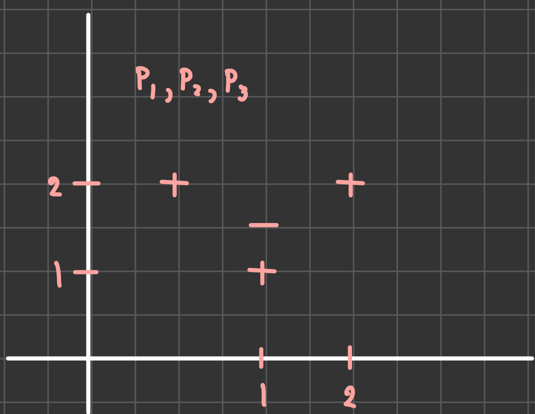
	x_1	x_2	P_1	P_2	target
e_1	0	0	1	0	0
e_2	0	1	1	1	1
e_3	1	0	1	1	1
e_4	1	1	0	1	0

ใช้ P1 จำแนก e_1, e_2, e_3 เป็นบวก   ใช้ P2 จำแนก e_2, e_3, e_4 เป็นบวก



INPUT LAYER + HIDDEN LAYER + OUTPUT LAYER = MLP Multi Layer Perceptron





$-$: (1,1)

$+$: (0,2), (1,0.5), (2,2)

27 Jan : Dichotomise and Capacity of a Perceptron & Number of Regions Seperated by Perceptrons

อยากรู้ว่าจาก p จุดใน n มิติ (gen position)
แยกได้โดยเส้นตรงกี่เส้น

Session overview (คลิปแรก)

Defination ของ
General Position

ex. กระจายตัวยังไงใน 2D (จุด 3 จุดไม่อยู่บนเส้นตรงเดียวกัน)
3D (จุด 4 จุดไม่อยู่ในระนาบเดียวกัน

จุด 4 จุดใน 2 มิติ

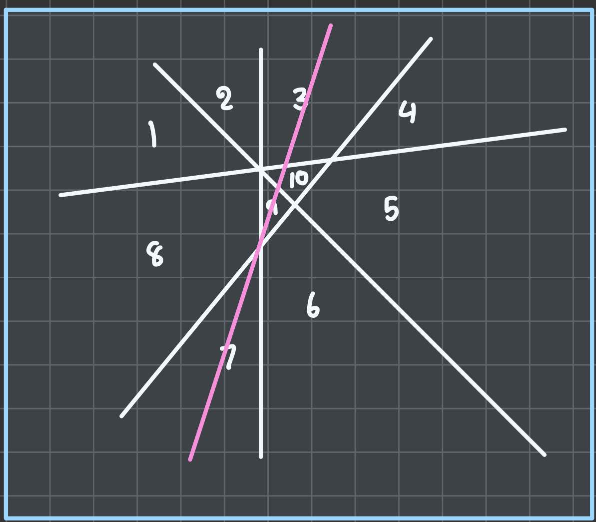
มีการแบ่ง (Dichotomines) $2^4 = 16$ แบบ

14 แบบ (linearly separable)

$\therefore \text{capacity} = \frac{14}{16}$

จุด 4 จุดใน 3 มิติ $\therefore \text{capacity} = \frac{16}{16}$

Session overview (คลิปสอง)

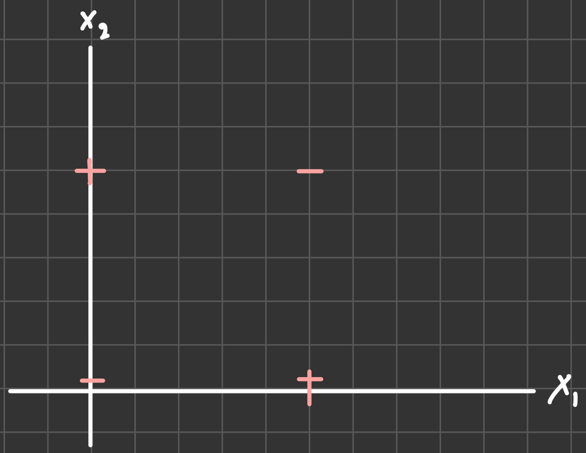


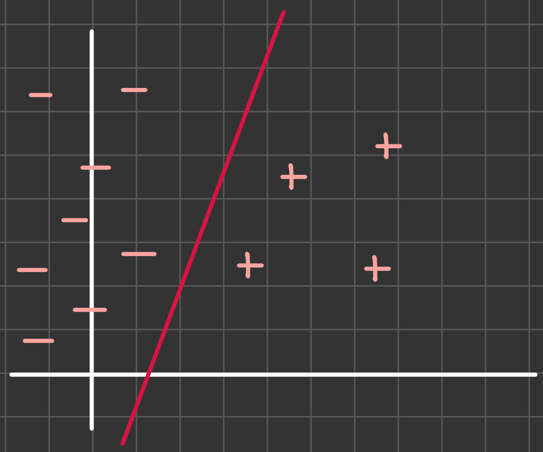
มีกี่ Region และถ้ามีเส้นใหม่มา อยากรู้ว่า
เราจะหา Region ยังไง
(จะเหลือ fomular ที่อยากให้เข้าใจ
การ prove และ used)

XOR (ไม่สามารถแบ่งได้ด้วย linearly Separable)

x_0	x_1	x_2	target (output)
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

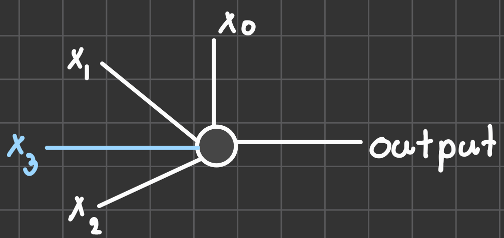
● = เหมือนกันออก 0





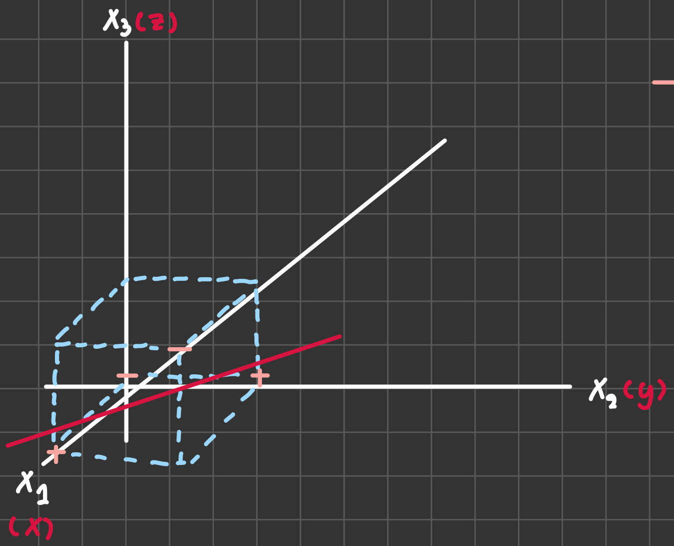
แล้วจะมีกี่case ที่แยก sample ได้

3D



$$\underset{x}{w_0x_0 + w_1x_1} + \underset{y}{w_2x_2} + \underset{z}{w_3x_3} > 0$$

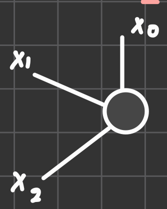
(มองเป็น x-y-z)

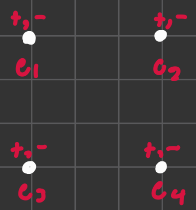


สมมติให้ x_3 เกิดจากการนำ $x_1 \cdot x_2$ จะได้

x	y	z	
x_1	x_2	x_3	target (output)
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

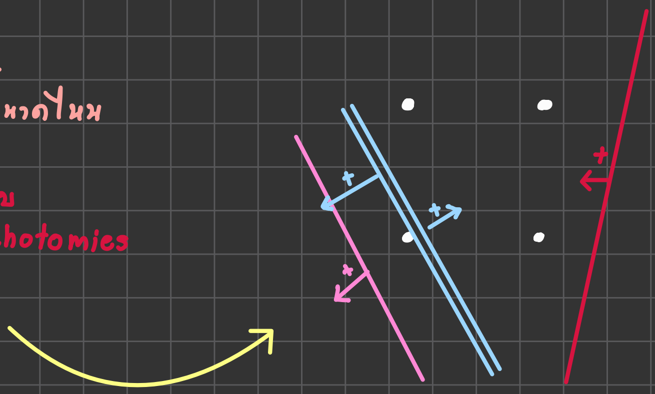
∴ การเพิ่มมิติทำให้เราแยกตัวอย่างได้ดียิ่งขึ้น
เลยมี concept ว่า perceptron มี potential
ในการแยก sample ได้ถึงขนาดไหน





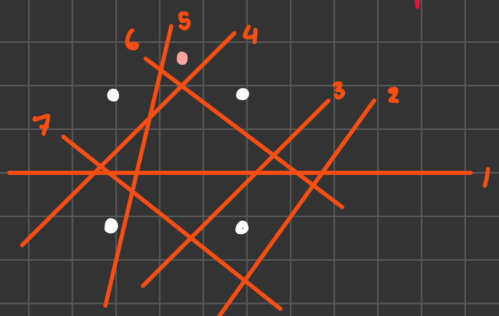
2^4 แบบ
16 dichotomies

มีจุดกระจายตัวใน
general position.
จุด p จุดใน n มิติ จะอยู่ใน gen pos
ก็ต่อเมื่อไม่มีจุดจำนวน n+1 จุด
วางใน hyperplane : n−1 มิติ

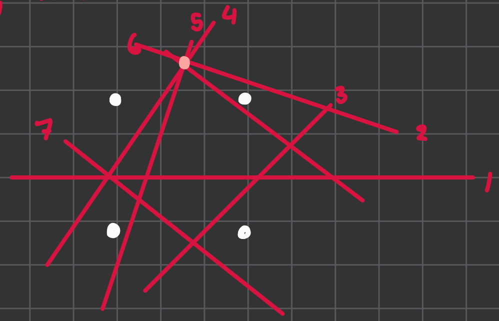


เราอยากรู้ว่ามีจุดจำนวนเท่าไหร่ โดยวาง
เส้นตรงลงไปใน space ที่จุดกระจายตัว
7 เส้น (14 แบบจาก 16 แบบที่สามารถ
แบ่งได้ด้วยเส้นตรงได้)

Suppose : datermine $L(p,n)$ • = จุดที่เพิ่มใหม่
↳ $p+1$ (เพราะมี 1 จุดเพิ่มมา)



Before



After.

L_p = จำนวนการแบ่งที่ลากผ่านจุดใหม่ได้

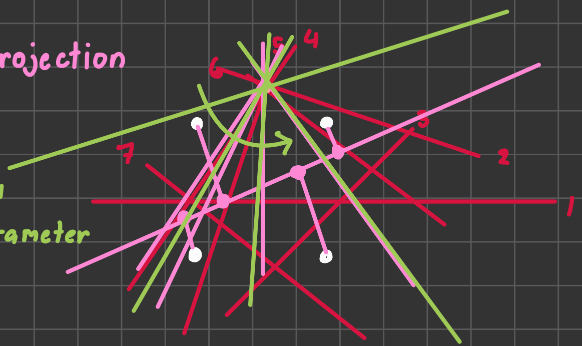
$L(p+1,n) = L(p,n) - L_p + 2L_p$ (3 จุด left / right)
จำนวนการแบ่งเดิมที่ลากผ่านจุดใหม่ไม่ได้

↓

$L(p+1,n) = L(p,n) + L_p$ (หน้าที่คือหา L_p)

1. — : ลาก projection ⊾

2. การปรับมุม
ได้แค่ 1 parameter



$L_p = L(p,n-1)$
$L(p+1,n) = L(p,n) + L(p,n-1)$

$L(1,0) = 2$
$L(1,1) = 2$
$L(1,2) = 2$
$\}$ ∴ $L(1,n) = 2 \quad n \geq 0$

$L(4,3) = L(3,3) + L(3,2)$

จุด 4 จุดใน space 3 มิติ $= p+1$; $p = 3$

$L(3,3) = L(2,3) + L(2,2)$

$L(3,2) = L(2,2) + L(2,1)$

$L(4,3) = L(2,3) + L(2,2) + L(2,2) + L(2,1)$

$= L(2,3) + 2L(2,2) + L(2,1)$

$L(2,3) = L(1,3) + L(1,2)$

$L(2,2) = L(1,2) + L(1,1)$

$L(2,1) = L(1,1) + L(1,0)$

$L(4,3) = L(1,3) + L(1,2) + 2L(1,2) + 2L(1,1) + L(1,1) + L(1\text{-}0)$

$= L(1,3) + 3L(1,2) + 3L(1,1) + L(1,0)$] สัมประสิทธิ์ ; เป็น 2 ตาม theory

$2 \quad 6 \quad 6 \quad 2$ = 16 แบบ

$\binom{3}{3} \quad \binom{3}{2} \quad \binom{3}{1} \quad \binom{3}{0}$

$1 \quad 3 \quad 3 \quad 3$

$$\binom{n}{r} = \frac{n!}{r!(n-r)!}$$

$$L(p,n) = \sum_{i=0}^{p-1} \binom{p-1}{i} \underbrace{L(1,n-i)}_{}$$

2 (ตาม theory)

$$L(p,n) = 2\sum_{i=0}^{p-1} \binom{p-1}{i}$$; บางที n น้อย / หมดก่อน

*

$$\boxed{= 2\sum_{i=0}^{\min(n,p-1)} \binom{p-1}{i}}$$

อยากได้ 4 จุดใน 2 มิติ จาก $L(p+1,n) = L(p,n) + L(p,n-1)$

$$L(4,2) = 2\sum_{i=0}^{\min(2,3)} \binom{3}{i}$$

$$= 2\left[\binom{3}{0} + \binom{3}{1} + \binom{3}{2}\right]$$

$$= 2[1 + 3 + 3]$$

$$= 2(7) = 14 \text{ แบบ}$$

อยากได้ 5 จุดใน 2 มิติ จาก $L(p+1,n) = L(p,n) + L(p,n-1)$

$$L(5,2) = 2\sum_{i=0}^{\min(2,4)} \binom{4}{i}$$

$$= 2\left[\binom{4}{0} + \binom{4}{1} + \binom{4}{2}\right]$$

$$= 2[1 + 4 + 6] = 2(11)$$

$$= 22 \text{ แบบ}$$

โดยสามารถหา prob ได้จาก $P(p,n) = \dfrac{L(p,n)}{2^p} = \begin{cases} 2^{1-p}\sum_{i=0}^{n}\binom{p-1}{i} \; ; & n<p-1,\; p>n+1 \\ 1 \; ; & p \le n+1 \end{cases}$ (แยกได้เป็น)

① ④

$$\frac{2\sum_{i=0}^{p-1}\binom{p-1}{i}}{2^p}$$

② ③

$$\boxed{\frac{2\left[\binom{p-1}{0}+\binom{p-1}{1}+\binom{p-1}{2}+\dots+\binom{p-1}{p-1}\right]}{2^p}} \rightarrow \frac{(2^{p-1})(2)}{2^p} = 1$$

For instance

$$\binom{4}{0}+\binom{4}{1}+\binom{4}{2}+\binom{4}{3}+\binom{4}{4}$$

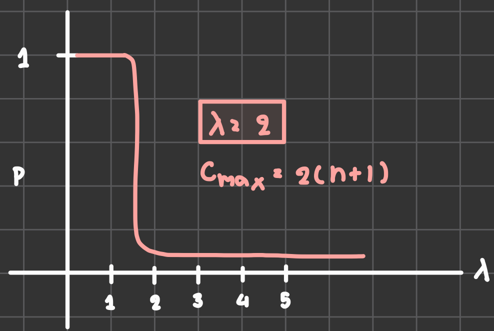
$$1 + 4 + 6 + 4 + 1 = 16 \longrightarrow 4^2$$
$$2^4$$

$$\binom{3}{0}+\binom{3}{1}+\binom{3}{2}+\binom{3}{3}$$

$$1 + 3 + 3 + 1 = 8 \longrightarrow 2^3$$

*

ให้จุดใดๆ $P = \lambda(n+1)$

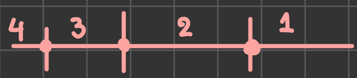


ถ้าใน Hidden Layer มี node จำนวนมาก จะหา region ที่เกิดขึ้นจาก Hidden Nodes ได้อย่างไร?

(Induction proof)

(1 เส้น อยากรู้ว่าตัดกี่เส้น)

พิจารณาจากกรณี 1 มิติ



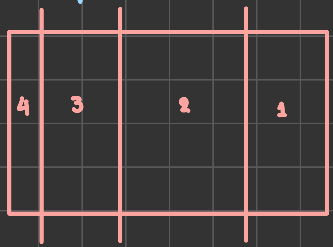
$\therefore n+1$

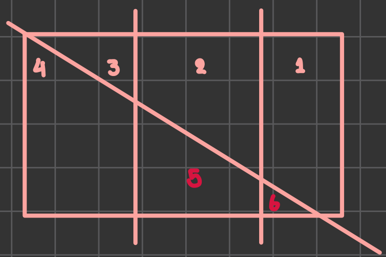
$3+1 = 4$

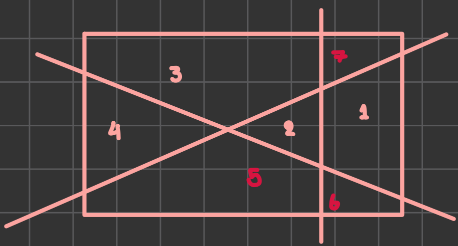
$\binom{3}{1} + \binom{3}{0}$

$3 \quad 1$

Suppose ยืดออกมาเป็น 2 มิติ เส้น —— สามารถบิดเพื่อเกิด region ใหม่ได้



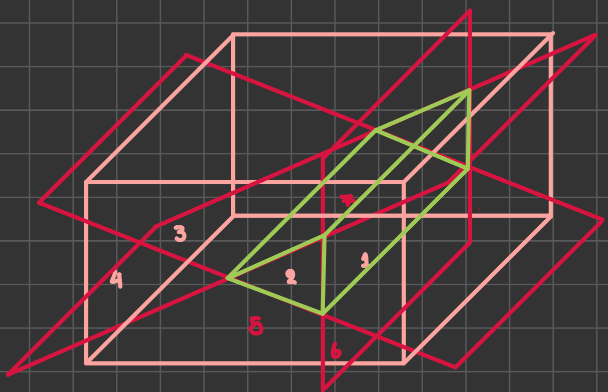


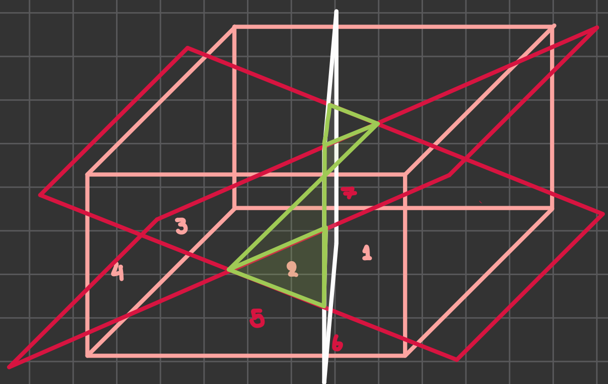


$3 + 3+1 = 7$

$\binom{3}{2} + \binom{3}{1} + \binom{3}{0}$

Suppose เพิ่มมาอีก 1 มิติ

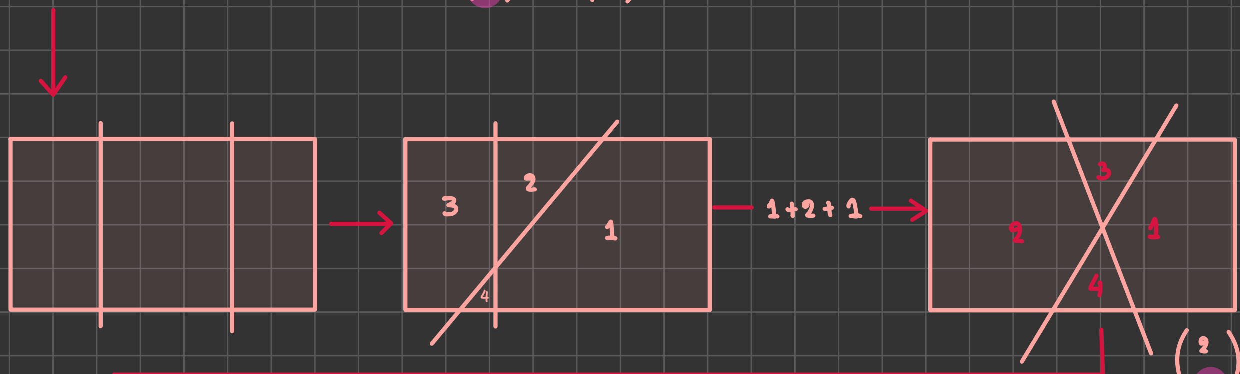




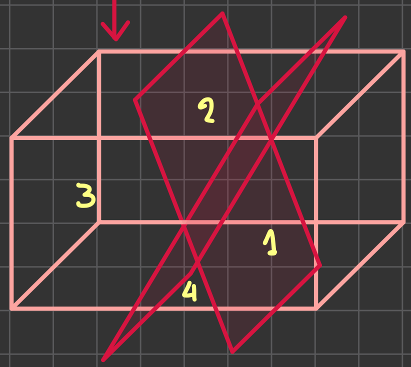
$1 + 3 + 3+1 = 8$

$\binom{3}{3} + \binom{3}{2} + \binom{3}{1} + \binom{3}{0}$

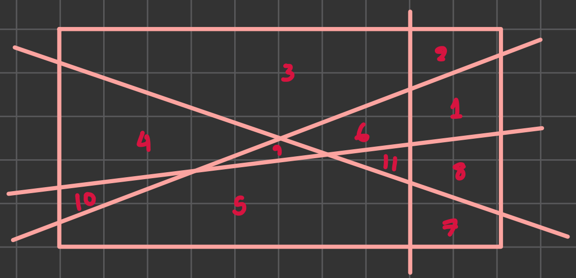
2+1 $\binom{2}{1} + \binom{2}{0}$



$\binom{2}{2} + \binom{2}{1} + \binom{2}{0}$



4+1 = 5



6+4+1 $\binom{4}{2} + \binom{4}{1} + \binom{4}{0}$

$\#$ region = $\sum_{i=0}^{n} \binom{n}{i}$ เมื่อ n คือจำนวนมิติ

20

3 Feb : TLN Training

ทบทวน !

General Position : จุด n+1 จุดจะต้องไม่วางตัวใน space (n-1) มิติ

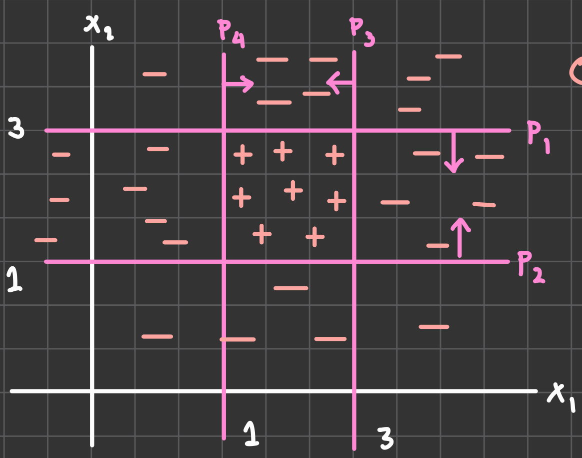
$L(p,n) = 2\sum_{i=0}^{\min(n,p-1)} \binom{p-1}{i}$: จำนวน Dichotomies ที่แบ่งโดยเส้นตรงหนึ่งเส้นได้

↓ จุด p จุด ใน n มิติ

ex. หาจุด p จุด ใน space 2 มิติ

Region : $\sum_{i=0}^{n}\binom{h}{i}$: เมื่อ n คือจำนวนมิติและ h คือจำนวน hyperplane

MLP : ถ้าเกิดว่ามี + กลาง - ล้อมรอบ อยากรู้ว่าโครงสร้างของ network ที่จำแนกชุดนี้ออกมาได้เป็นอะไร ?



Q : ให้เขียน poceptron structure weight ที่ classifier sample (+,-) โดยใช้ TLN (ในคลิป MLP Specific Area)

; ต้องการเส้นตัด 4 เส้น ($P_1 - P_4$) และสมการ NN

$W_0 + W_1X_1 + W_2X_2 > 0$

P_1 : $x_2 < 3$

$-x_2 + 3 > 0$ (ย้าย x_2 มาให้ตรงตามสมการ nn) ; $w_2 = -1$ (ดูเลขสัมประสิทธิ์หน้า x_2) ส่วน $w_1 = 0$ (เพราะไม่มี x_1)

P_2 : $x_2 > 1$

$-1 + x_2 > 0$

P_3 : $x_1 < 3$

$-x_1 + 3 > 0$

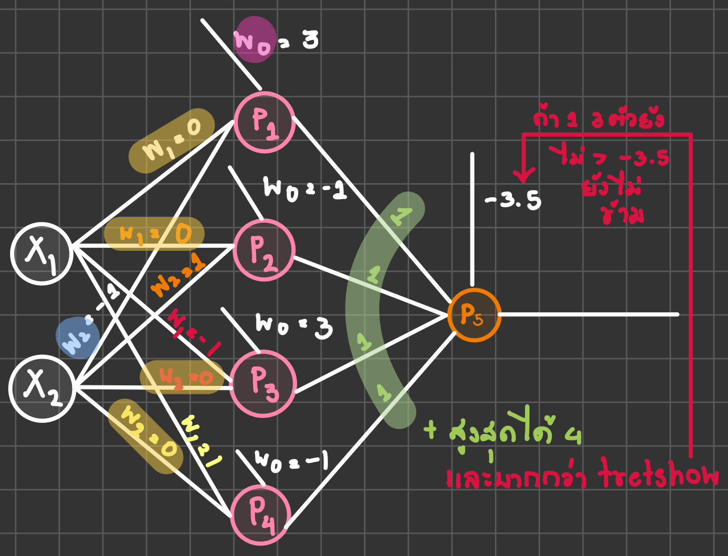
P_4 : $x_1 > 1$

$-1 + x_1 > 0$

↓ ได้ weight ละ

P_1	P_2	P_3	P_4	
T	T	T	T	
1	1	1	1	→ 1

AND 4 INPUT



Wieght = 0 ตัดทิ้งได้

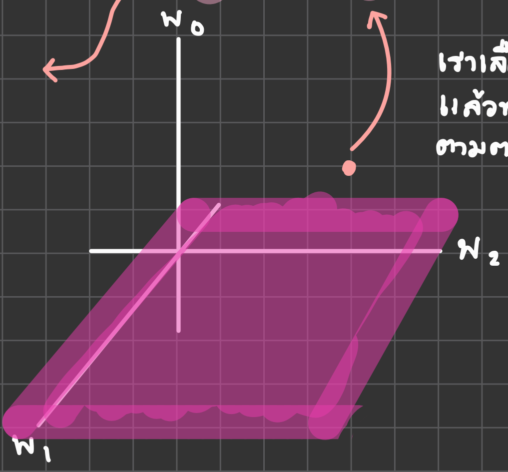
Train TLN

AND TABLE

x_1	x_2	target
0	0	−
0	1	−
1	0	−
1	1	+

สมการที่อยากได้

$w_0 + w_1x_1 + w_2x_2 > 0$

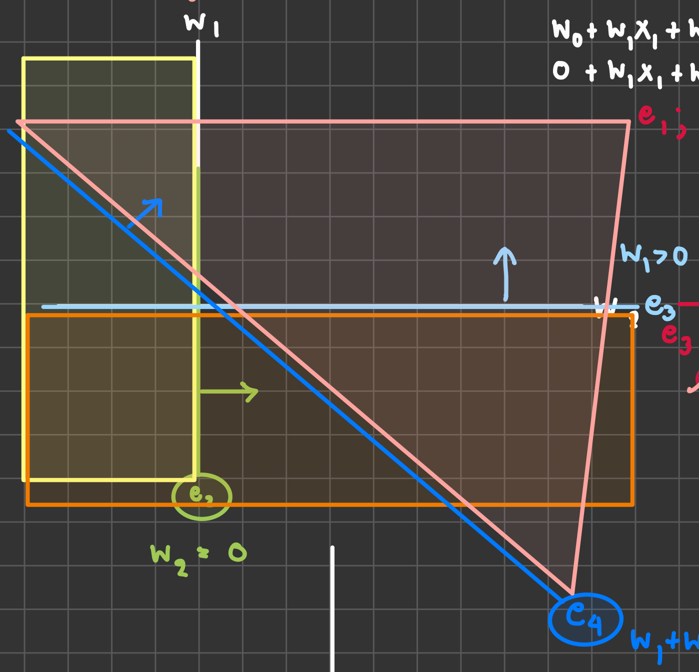


เราเลือกมาจุดนึงใน 3D space แล้วทำให้ทุก w classifier ตามตารางทั้งหมด

ถ้าให้ $w_0 = 0$

$w_0 + w_1x_1 + w_2x_2 > 0$

$0 + w_1x_1 + w_2x_2 > 0$



e_1 ; $x_1 = 0$, $x_2 = 0$

0 > 0 FALSE ทั้งแถบ

จะเอา w_1, w_2 ตรงไหนก็ได้

e_2 ; $x_1 = 0$, $x_2 = 1$

$\cancel{0} + \cancel{w_1(0)} + w_2(1) > 0$

$\boxed{w_2 > 0}$

e_3 ; $x_1 = 1$, $x_2 = 0$

$\cancel{0} + w_1(1) + \cancel{w_2(0)} > 0$

$\boxed{w_1 > 0}$

e_4 ; $x_1 = 1$, $x_2 = 1$

$0 + w_1(1) + w_2(1) > 0$

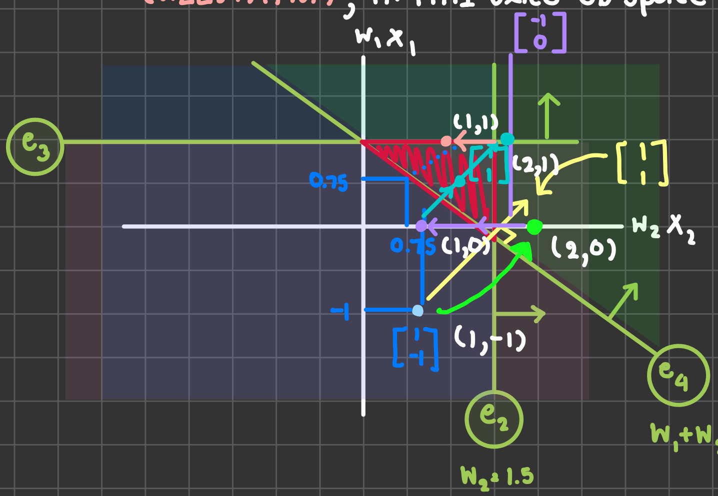
$\boxed{w_1 + w_2 > 0}$

ไอเดีย ! : อยากได้ region ร่วมกันและการปรับ weight

∴ ไม่มี region ร่วมกัน

$w_0 = -1.5$

(ลบอะไรก็ได้) ; ทำ MRI slice 3D space เป็นขนมปังแผ่นๆ แล้วหยิบแผ่น $w_0 = -1.5$



e_1 ; $x_1 = 0$, $x_2 = 0$

$-1.5 + w_1(0) + w_2(0) > 0$

$-1.5 > 0$

ALL FALSE !

e_2 ; $x_1 = 0$, $x_2 = 1$

$-1.5 + w_1(0) + w_2(1) > 0$

$-1.5 + w_2 > 0$

e_3 ; $x_1 = 1$, $x_2 = 0$

$-1.5 + w_1(1) + w_2(0) > 0$

$-1.5 + w_1 > 0$

e_4 ; $x_1 = 1$, $x_2 = 1$

$-1.5 + w_1(1) + w_2(1) > 0$

$-1.5 + w_1 + w_2 > 0$

$w_1 + w_2 > 1.5$

∴ หยิบจุดใดๆ ใน ◺ จะต้องจำแนก sample ได้ถูกต้องทั้งหมด !

สมมติตอนแรกสุ่มจุดสีฟ้า (.) มา classifier e_1, e_2, e_3 ถูก

How to ขยับ จุดสีฟ้า (.) มาอยู่เหนือเส้นได้

ANS + vector $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ จากตำแหน่งเดิม $\begin{bmatrix} 1 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$

ตก e_1, e_3, e_4

อยากให้วกกลับมา $\begin{bmatrix} -1 \\ 0 \end{bmatrix} + \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \longrightarrow$ + vector $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$

ตก e_1, e_2, e_3

จะเห็นว่าเวลา + vector อาจเลย ทำให้มี 2 กรณี :

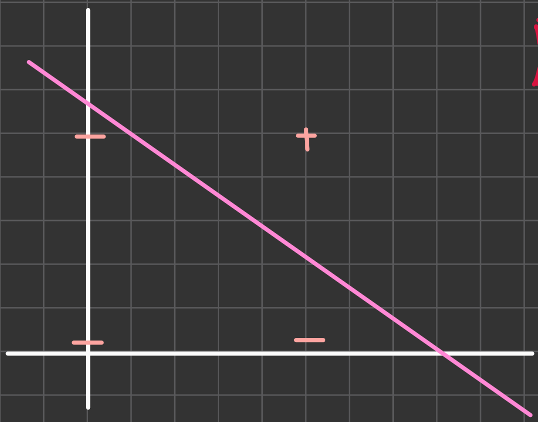
$\vec{w} = \vec{w} + \Delta w$

$$\Delta w = \begin{cases} \eta\vec{x} & \text{ถ้าเป็น ตย.บวก และ } \vec{w}\cdot\vec{x} < 0 \\ -\eta\vec{x} & \text{ถ้าเป็น ตย.ลบ และ } \vec{w}\cdot\vec{x} > 0 \end{cases}$$

- คูณด้วย learning rate เพื่อไม่ให้ปรับค่าไปไกลเหมือนตัวอย่าง

ค่าคงที่น้อยๆ.

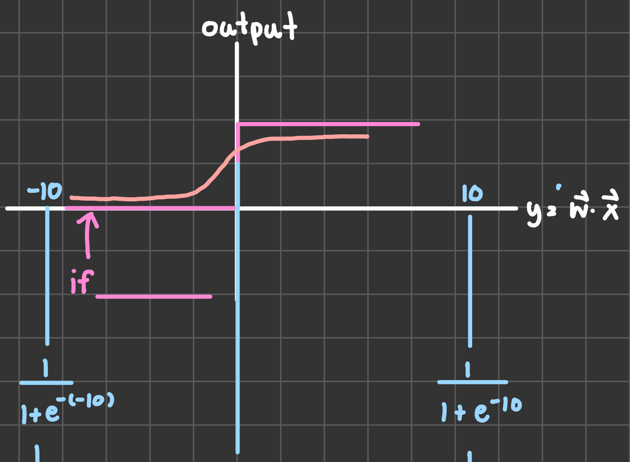
10 Feb : Signal Function



$\vec{W} = \vec{W} + \Delta W$

$$\Delta\vec{W} = \begin{cases} \eta\vec{x} & \text{if } \vec{W}\cdot\vec{x} < 0 \text{ and positive} \\ -\eta\vec{x} & \text{if } \vec{W}\cdot\vec{x} > 0 \text{ and negative} \end{cases}$$

Signal Function TLN



$\frac{1}{1+e^{-(-10)}}$ $\frac{1}{1+e^{-10}}$

$\frac{1}{1+e^{10}}$ $\frac{1}{1+e^{-0}}$ $\frac{1}{1+e^{-10}}$

$\frac{1}{1+\infty}$ (เข้าใกล้0) $\frac{1}{1+1} = 0.5$ $\frac{1}{1.00\sim}$ (เกือบ 1)

มากกว่า 0 ออก 1
น้อยกว่า 0 ออก 0

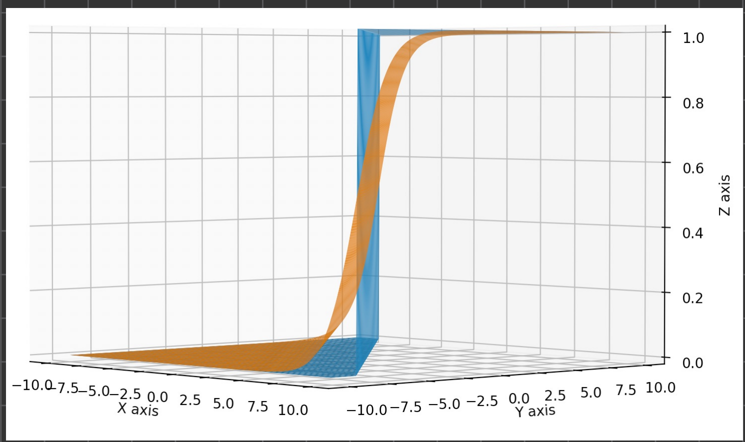
↓

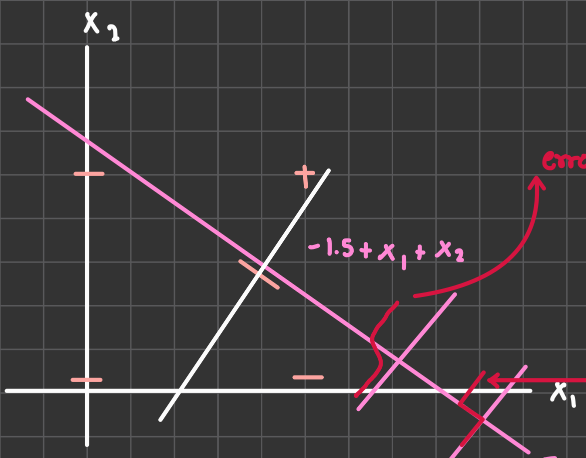
ถ้าคำนวณเยอะๆจะเหนื่อย/เลยขอ
ใช้ Sigmoid $\sigma(y) = \frac{1}{1+e^{-y}}$

$\sigma(\vec{W}\cdot\vec{x}) = \frac{1}{1+e^{-\vec{W}\cdot\vec{x}}}$

↓

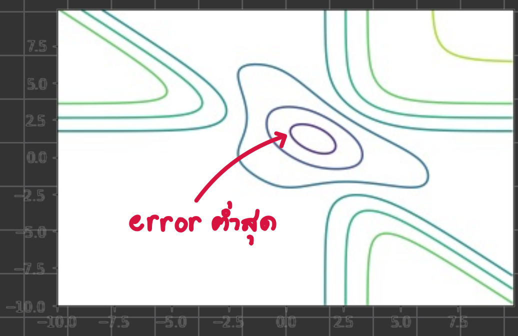
ทำหน้าที่แบบเดียวกัน แต่ sigmoid
ไม่ต้อง if !

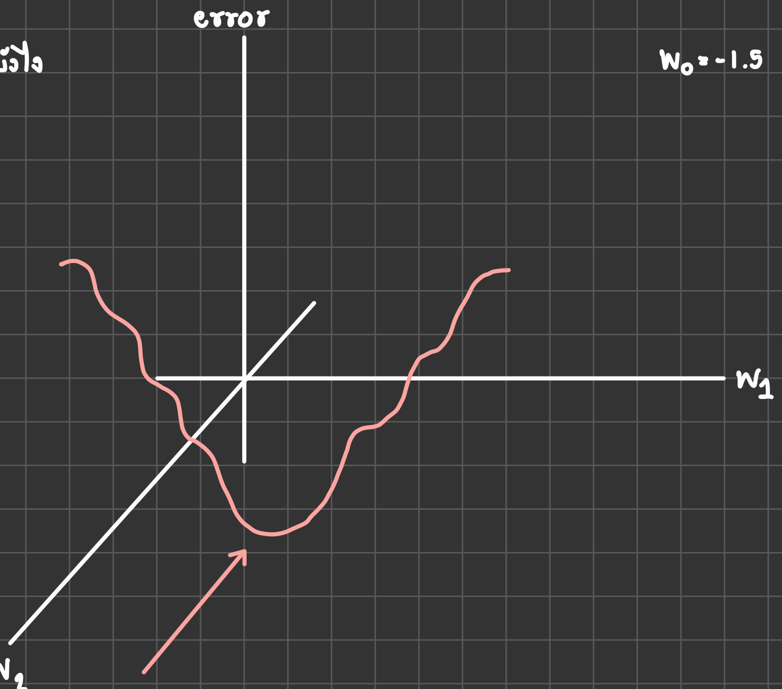




ถ้าเกิดใช้ Sigmoid error surface จะเป็นยังไง

$W_0 = -1.5$





$e = \frac{1}{2}(\text{target}-\text{output})^2$

ถ้า error อยู่ใน form นี้ ชอบมี 1/2 ไว้ข้างหน้า

$\nabla e = \frac{\partial}{\partial w}\cdot\frac{1}{2}(\text{target}-\text{output})^2$

$= \frac{1}{2}\frac{\partial}{\partial w}(\text{target}-\text{output})^2$

$= \cancel{\frac{1}{2}}(\cancel{2})(t-o)\frac{\partial}{\partial w}(t-o)$

target = constant

$= (t-o)\frac{\partial}{\partial w}(\cancel{t}-o)$

$= (t-o)\frac{\partial}{\partial w}(-o)$

↳ output = sigmoid

$= -(t-o)\frac{\partial}{\partial w}\left(\frac{1}{1+e^{-w\cdot x}}\right)$

$\partial(\vec{w}\cdot\vec{x})$ และถ้า diff เทียบตัวมันเองจะได้

$\frac{\partial\sigma(\vec{w}\cdot\vec{x})}{\partial\vec{w}\cdot\vec{x}} = \sigma(\vec{w}\cdot\vec{x})(1-\sigma(\vec{w}\cdot\vec{x}))$

$\frac{\partial}{\partial u}\frac{1}{1+e^{u}} = \frac{\partial}{\partial u}(1+e^{-u})^{-1}$

$= -\frac{1}{(1+e^{-u})^2}\frac{\partial}{\partial u}e^{-u}$

$= \boxed{+\frac{e^{-u}}{(1+e^{-u})^2}} \rightarrow \left(\frac{1}{1+e^{-u}}\right)\left(\frac{1+e^{-u}-1}{1+e^{-w}}\right) = \frac{e^{-u}}{(1+e^{-u})^2}$

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

chaining

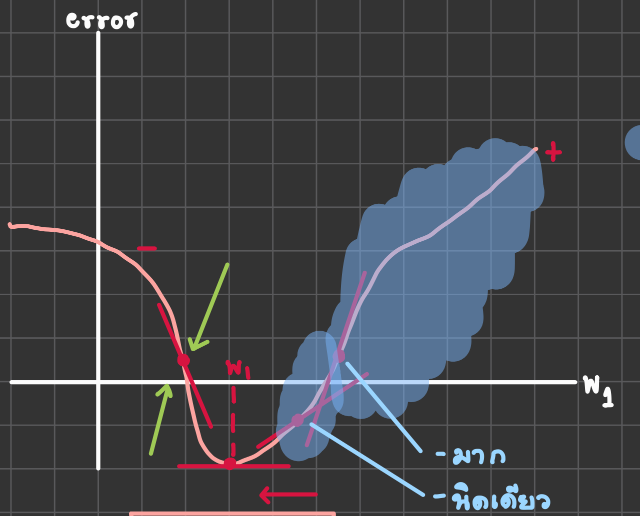
$\frac{\partial\sigma(\vec{w}\cdot\vec{x})}{\partial w} = \frac{\partial\sigma(\vec{w}\cdot\vec{x})}{\partial\vec{w}\cdot\vec{x}}\cdot\frac{\partial\vec{w}\cdot\vec{x}}{\partial\vec{w}}$

$= \sigma(\vec{w}\cdot\vec{x})(1-\sigma(\vec{w}\cdot\vec{x}))\cdot\vec{x}$

เปลี่ยนเป็น output

$= -o(1-o)\vec{x}$ (เป็นความชัน)

มองจาก W_2 สนใจเฉพาะ W_1



ความชันเป็น (+) มาก ๆ ลด w_1 ให้ลด

$w_1 = w_1 - \eta\,$ความชัน

$w_1 = w_1 - \eta\,$ความชัน (ความชันติดลบ ; ลบความชันทำให้ค่าของ w_1 เพิ่มขึ้น)

$-\eta\cdot$ ความชันตอนปรับจะได้ขยับน้อยลง

∴ ไม่ว่ามองจาก w ไหนก็เป็นแบบนี้หมด การปรับ weight ปรับเป็น weight ของ vector ได้เลย !

$\vec{w} = \vec{w} - \eta\nabla e$ (ความชัน, gradient ของ error surface)

$\vec{w} = \vec{w} - \eta(t-o)(-o)(1-o)\vec{x} \rightarrow \vec{w} = \vec{w} + \eta(t-o)(o)(1-o)\vec{x}$

ความชันของ Sigmoid Function (จากคลิป)

$$\sigma(y) = \frac{1}{1+e^{-y}} \qquad y = w_i x_i = \vec{w}\cdot\vec{x}$$

เราจะหาdiff sigmoid เทียบกับ y

$$\frac{d}{dy}\sigma(y) = \sigma(y)(1-\sigma(y))$$; ความชันเทียบ y

หา error

$$error = \frac{1}{2}(t-o)^2$$

diff error เทียบ w

$$\frac{de}{d\vec{w}} = \frac{1}{2}\frac{d}{d\vec{w}}(t-o)^2$$

$$= \frac{1}{\cancel{2}}\cancel{2}(t-o)\frac{d}{d\vec{w}}(t-o)$$

ค่าคงที่

$$= (t-o)\left(-\frac{do}{d\vec{w}}\right)$$

$$= -(t-o)\frac{d}{d\vec{w}}\sigma(\vec{w}\cdot\vec{x})$$

$$= -(t-o)\,\sigma(\vec{w}\cdot\vec{x})(1-\sigma(\vec{w}\cdot\vec{x}))\frac{d}{d\vec{w}}\vec{w}\cdot\vec{x}$$

$$= -(t-o)\,\sigma(\vec{w}\cdot\vec{x})(1-\sigma(\vec{w}\cdot\vec{x}))\,\vec{x}$$

$$= -(t-o)(o)(1-o)\,\vec{x}$$

$$\vec{w} \longleftarrow \vec{w} + \Delta\vec{w}$$

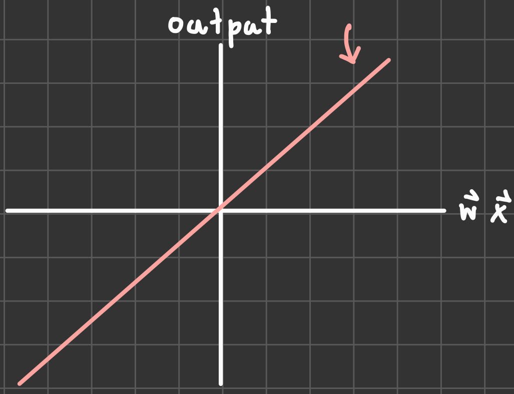
$$\Delta\vec{w} = \eta(t-o)(o)(1-o)\,\vec{x}$$

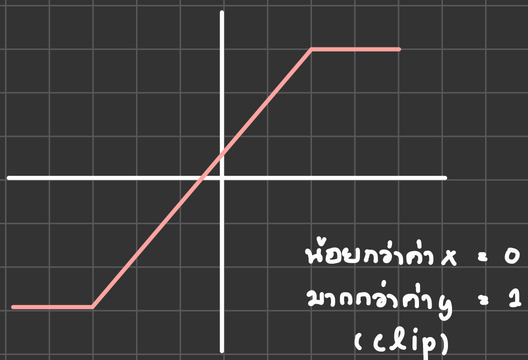
Sigmoid node

$\vec{w} = \vec{w} + \Delta\vec{w}$

$$\Delta\vec{w} = \eta\underbrace{(t-o)}_{\text{เกิดจากการคำนวณ error}}\underbrace{(o)(1-o)\vec{x}}_{\text{เกิดจากความชัน}}$$

Linear node





$$
\begin{aligned}
\text{output} &= \vec{w}\cdot\vec{x} \\
\text{error} &= \frac{1}{2}\times(t-o)^2 \\
\vec{w} &= \vec{w} - \eta\nabla e \\
\nabla e &= \frac{\partial}{\partial\vec{w}}\left(\frac{1}{2}\right)(t-o)^2 \\
&= \frac{1}{\cancel{2}}(\cancel{2})(t-o)\frac{\partial}{\partial\vec{w}}(\cancelto{0}{t}-o) \\
&= -(t-o)\frac{\partial o}{\partial\vec{w}} \\
&= -(t-o)\cancel{\frac{\partial\vec{w}}{\partial\vec{w}}}\cdot\vec{x} \\
&= -(t-o)\vec{x}
\end{aligned}
$$

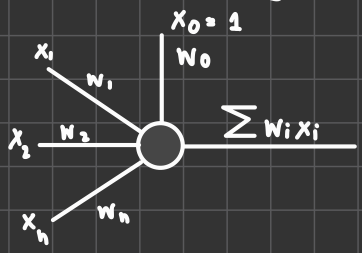
$\vec{w} = \vec{w} + \Delta\vec{w}$

$\Delta\vec{w} = \eta(t-o)\vec{x}$

ข้อสอบ: ให้ activation func มา
อยากดูสมการในการปรับ weigth

17 Feb : Weiner Solution / Gaussian

Linear node : $S(y) = y$



$\vec{w} \leftarrow \vec{w} + \Delta w$ ใหม่ เดิม

$\Delta w = \eta(t-o)x \longrightarrow$ train ทีละตัว ในหนส. = d_k (target ตัว k ใดๆ)

t_k

$\varepsilon_k = \frac{1}{2}(t_k - \vec{x}_k^T \vec{w}_k)^2 = \frac{1}{2} e_k^2$

$= \frac{1}{2}(t_k^2 - 2t_k \vec{x}_k^T \vec{w}_k + \underbrace{\vec{w}_k^T \vec{x}_k \vec{x}_k^T \vec{w}_k}_{\text{metrix}^2})$

transpot เพราะอยากให้เป็นสเกล่าร์

expected val

$\varepsilon = E[\varepsilon_k] = \frac{1}{2} E[t_k^2] - E[t_k x_k^T] \vec{w}_k + \frac{1}{2} \vec{w}_k E[\vec{x}_k x_k^T] \vec{w}_k$

สเกล่าร์ แทนด้วย P แทนด้วย R

column vector

$x_1, x_2 = 2 \times 1$

$\varepsilon = \frac{1}{2} E[t_k^2] - P\vec{w}_k + \frac{1}{2} \vec{w}^T R w_k$

หา error surface diff เทียบ w (หา gradient)

$\frac{\partial \varepsilon}{\partial \vec{w}} = -P + RW = 0$

$W = PR^{-1}$