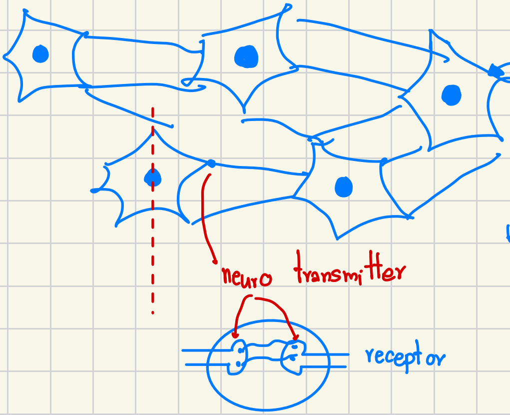
Neural Networks → โครงข่ายประสาท

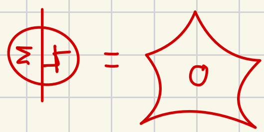


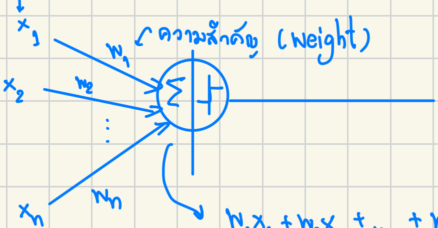
neurons → เซลล์ประสาท

$10^{10} - 10^{11}$ neurons

แต่ละ neuron เชื่อมกัน 10^{4-5} connections / neuron

แต่ละเส้นเชื่อมมี weight ไม่เท่ากัน และมีการ form สมองในรูปแบบใหม่ ๆ ตลอด





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

$> b$ แปลว่า ถ้าสัญญาณขาเข้ามากพอ จะทำให้ output ถูก activate

$$\text{output} = \begin{cases} 1 & \text{if } w_1x_1 + w_2x_2 + \dots + w_nx_n > b \\ 0 & \text{otherwise} \end{cases}$$

→ ค่านี้ถูกใช้ใน SNA เช่น ค่ามากพอที่เราจะกด share

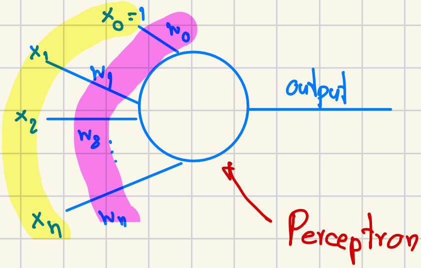
$$w_1x_1 + w_2x_2 + \dots + w_nx_n > \underset{\text{bias}}{b}$$

$$-b + w_1x_1 + w_2x_2 + \dots + w_nx_n > 0$$

$$w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n >$$

$$w_0x_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n > 0 \quad ; x_0 = 1$$

$$\sum_{i=0}^{n} w_ix_i > 0$$

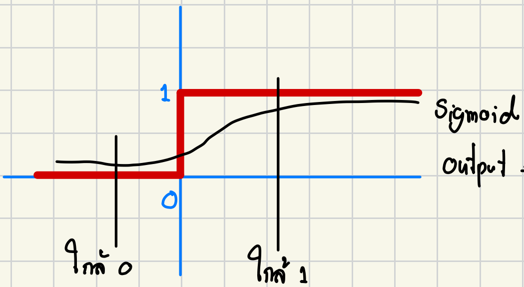


$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \cdot \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$$

$$\vec{w} \cdot \vec{x} = \sum_{i=0}^{n} w_i x_i$$

$$\left.\text{output} = \begin{cases} 1 & \text{if } \vec{w} \cdot x > 0 \\ 0 & \text{otherwise} \end{cases}\right\} \text{binary perceptron}$$

$$\left.\text{output} = \begin{cases} 1 & \text{if } \vec{w} \cdot \vec{x} > 0 \\ -1 & \text{otherwise} \end{cases}\right\} \text{bipolar perceptron}$$



$$\text{output} = \text{Sigmoid}(\vec{w} \cdot \vec{x})$$

$$= \frac{1}{1 + e^{-\vec{w} \cdot \vec{x}}}$$

$$\text{Sigmoid}(y) = \frac{1}{1 + e^{-y}}$$

$$\frac{\partial}{\partial y} \text{Sigmoid}(y) = \text{Sigmoid}(y)(1 - \text{Sigmoid}(y))$$

$$= \frac{d}{du} u^{-1} \cdot \frac{du}{dy}$$

$$= -(u^{-2}) \cdot \frac{d}{dy}(1 - e^{-y})$$

$$= -(1+e^{-y})^{-2}(e^{-y})\frac{d}{dy}(-y)$$

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

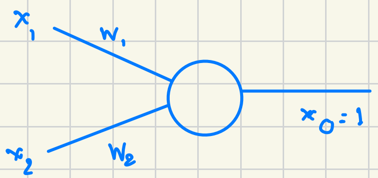
$$= \frac{e^{-y}}{(1+e^{-y})^2}$$

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

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

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

Geometry of Perceptron



$$w_1x_1 + w_2x_2 > b$$

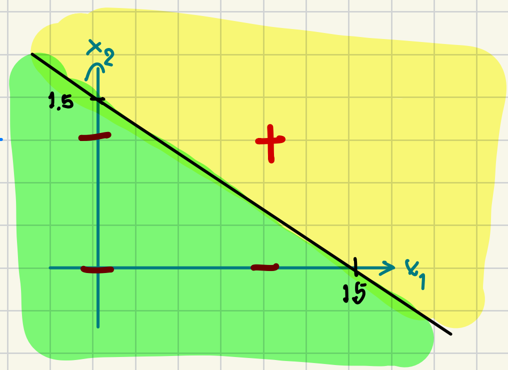
$$-b + w_1x_1 + w_2x_2 > 0$$

$$w_0x_0 + w_1x_1 + w_2x_2 > 0$$

AND Table

x_1	x_2	target
T	T	T
T	F	F
F	T	F
F	F	F

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



$x_1 = 1, x_2 = 1$ $\quad -1.5 + 1\cancel{x_1}^{1} + 1\cancel{x_2}^{1} > 0$

$0.5 > 0$

$x_1 = 1, x_2 = 0$ $\quad -0.5 > 0$

$x_1 = 0, x_2 = 1$ $\quad -0.5 > 0$

$x_1 = 0, x_2 = 0$ $\quad -0.5 > 0$

$$w_0 + w_1x_1 + w_2x_2 > 0$$

$c + ax + by > 0$ สมการเส้นตรง

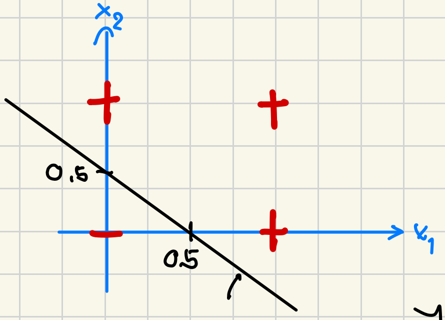
$$c + ax + by = 0$$

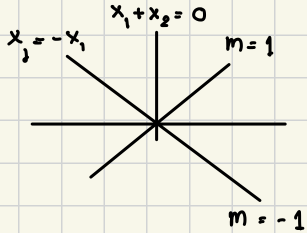
$$y = -\frac{ax}{b} - \frac{c}{b}$$

$$y = mx + c$$

OR Table

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





$x_1 + x_2 = 0.5$ แยกได้ 2 แบบคือ

$-0.5 + x_1 + x_2 = 0$	$0.5 - x_1 - x_2 = 0$
$-0.5 + x_1 + x_2 > 0$	$0.5 - x_1 - x_2 > 0$

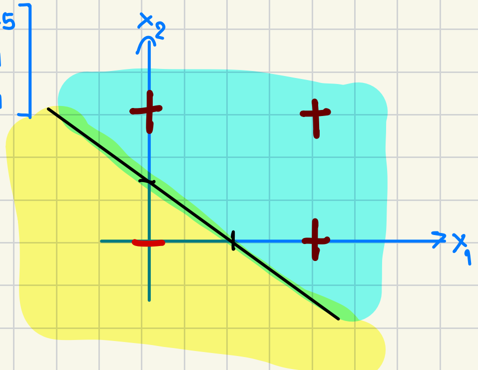
$x_1 = 0, x_2 = 0$: $-0.5 + 0 + 0 > 0$ → $-0.5 > 0$ ⊖ | $0.5 - 0 - 0 > 0$ → $0.5 > 0$ ⊕

$x_1 = 0, x_2 = 1$: $-0.5 + 0 + 1 > 0$ → $0.5 > 0$ ⊕ | $0.5 - 0 - 1 > 0$ → $-0.5 > 0$ ⊖

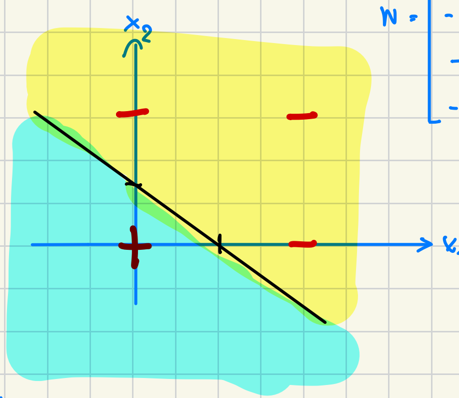
$x_1 = 1, x_2 = 0$: $-0.5 + 1 + 0 > 0$ → $0.5 > 0$ ⊕ | $0.5 - 1 - 0 > 0$ → $-0.5 > 0$ ⊖

$x_1 = 1, x_2 = 1$: $-0.5 + 1 + 1 > 0$ → $1.5 > 0$ ⊕ | $-0.5 - 1 - 1 > 0$ → $-1.5 > 0$ ⊖

$$W = \begin{bmatrix} -0.5 \\ 1 \\ 1 \end{bmatrix}$$



$$W = \begin{bmatrix} -0.5 \\ -1 \\ -1 \end{bmatrix}$$

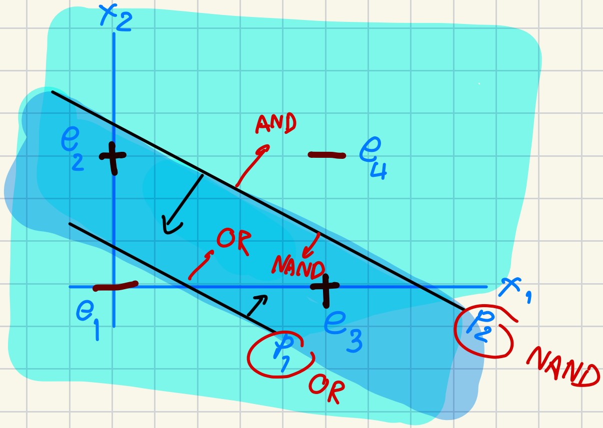


เลือกสมการฝั่งนี้ เพราะตรงกับ OR Table

Linearly Seperable

XOR TABLE

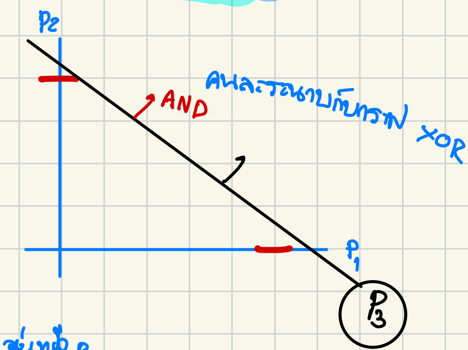
	x_1	x_2	target
e_1	0	0	−
e_2	0	1	+
e_3	1	0	+
e_4	1	1	−

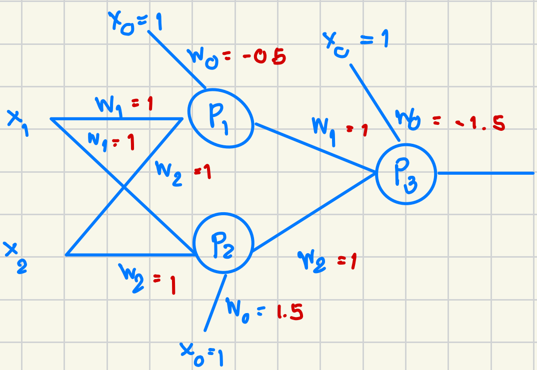


	x_1	x_2	P_1	P_2	target
e_1	0	0	0	1	−
e_2	0	1	1	1	+
e_3	1	0	1	1	+
e_4	1	1	1	0	−

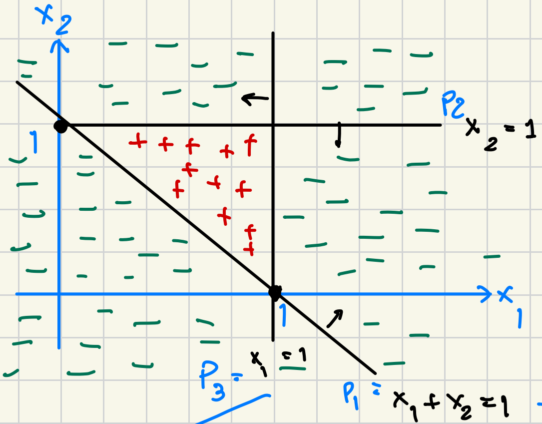
$e_1 = 0$ เพราะอยู่ใต้เส้น p_1

$e_4 = 0$ เพราะอยู่เหนือ p_2





Multilayer Perceptron (MLP)



$\boxed{x_2 < 1}$

$\boxed{1 - x_2 > 0}$

$\boxed{x_1 < 1}$

$\boxed{1 - x_1 > 0}$

$x_1 + x_2 > 1$

$1 + 1 > 1$

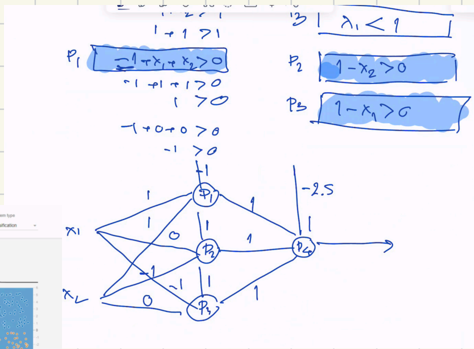
$-1 + x_1 + x_2 > 0$

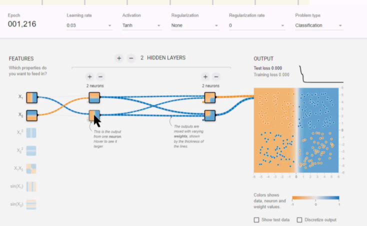
$\boxed{-1 + 1 + 1 > 0}$ P_1

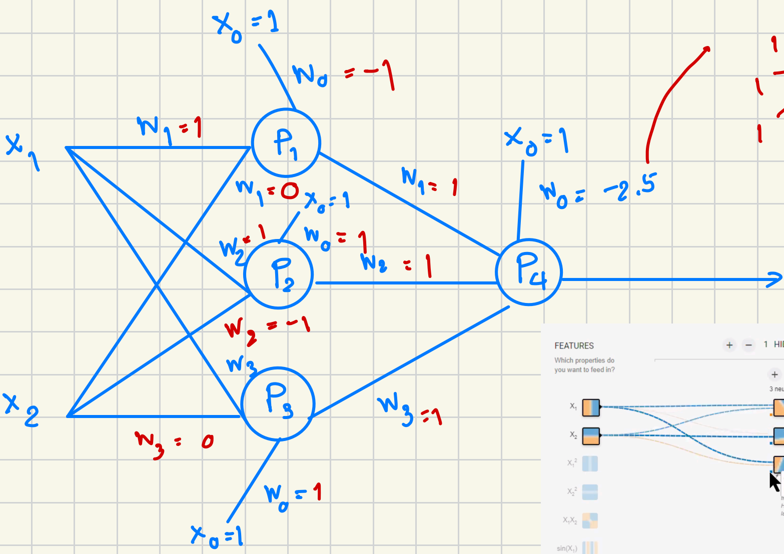
$1 > 0$ ✓

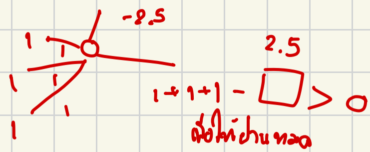
$0,0$ $\quad -1 + 1 + 1 > 0$

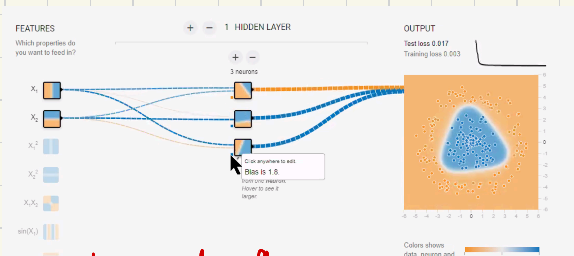
$-1 > 0$ ✗





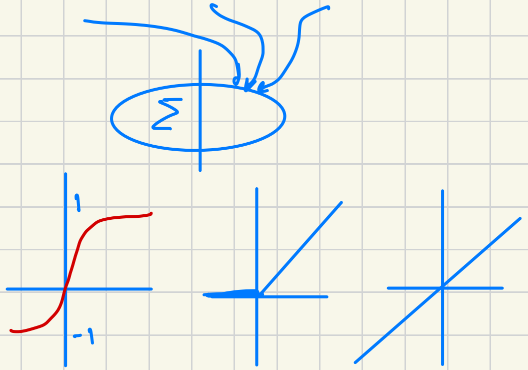






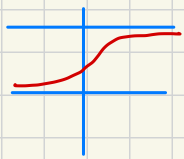
playground.tensorflow.org

tanh ReLU Linear

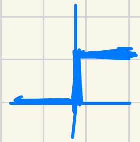


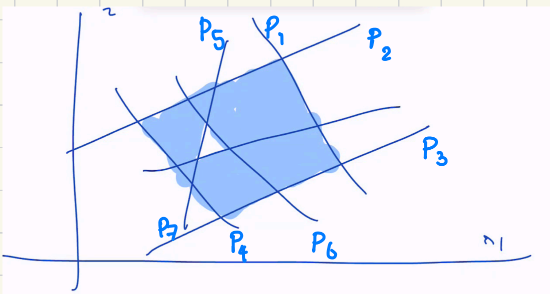
Sigmoid

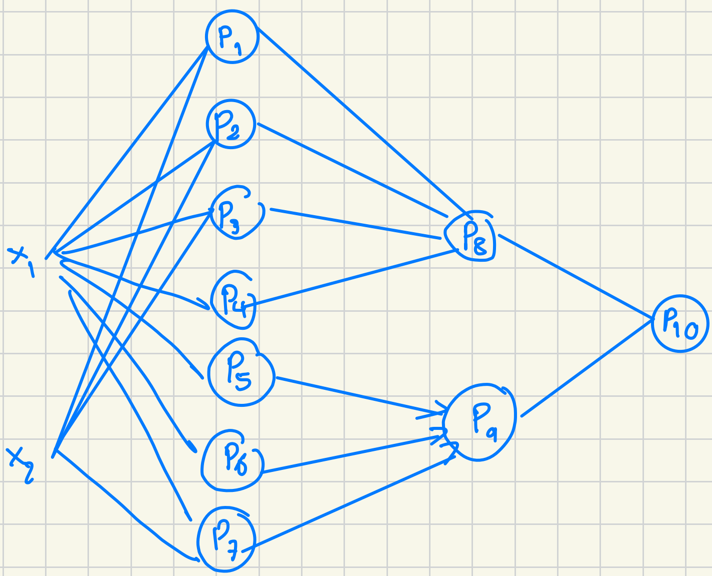
$$\frac{1}{1+e^{-\vec{w}\cdot x}}$$



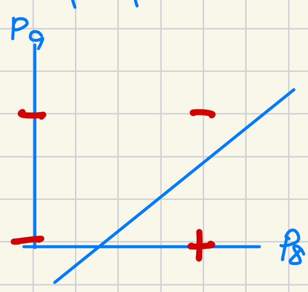
TLN





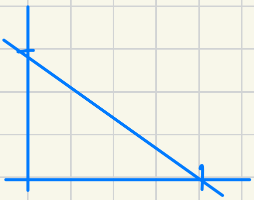


P_8	P_9	
1	0	+
0	0	−
0	1	−
1	1	−



การ Training ปรับ weight

① Binary + Bipolar



if ...

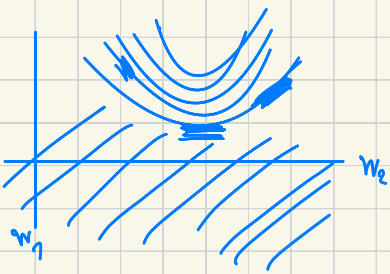
otherwise ..

$$\boxed{\begin{array}{llll} \text{if } x \text{ เป็น } + & \text{แต่ } \vec{w}\cdot\vec{x} < 0 & ; & \vec{w} = \vec{w} + \eta\vec{x} \\ \text{else } x \text{ เป็น } - & \text{แต่ } \vec{w}\cdot\vec{x} > 0 & ; & \vec{w} = \vec{w} - \eta\vec{x} \end{array}}$$

บางกรณีก็หารขนาดของ $\vec{x}$

$$\eta \frac{\vec{x}}{\|\vec{x}\|}$$

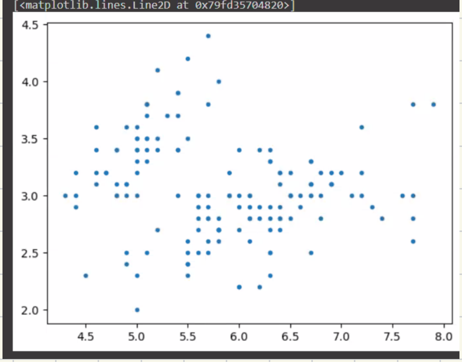
② Gradient Descent



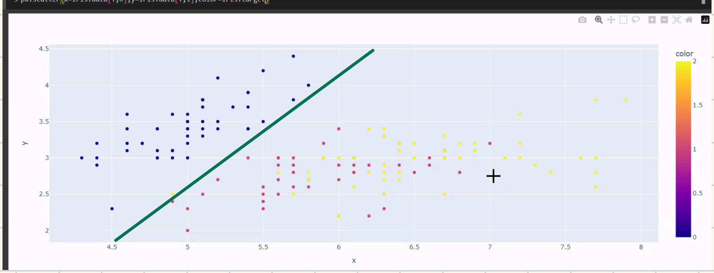
Colaboratory

```
import matplotlib.pyplot as plt

plt.plot(iris.data[:,0],iris.data[:,1],'.')
```

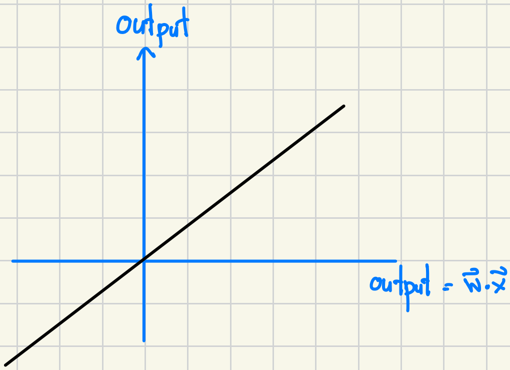
```
[<matplotlib.lines.Line2D at 0x79fd35704820>]
```




```
import plotly.express as px

px.scatter(x=iris.data[:,0],y=iris.data[:,1],color=iris.target)
```



Training a Linear Node

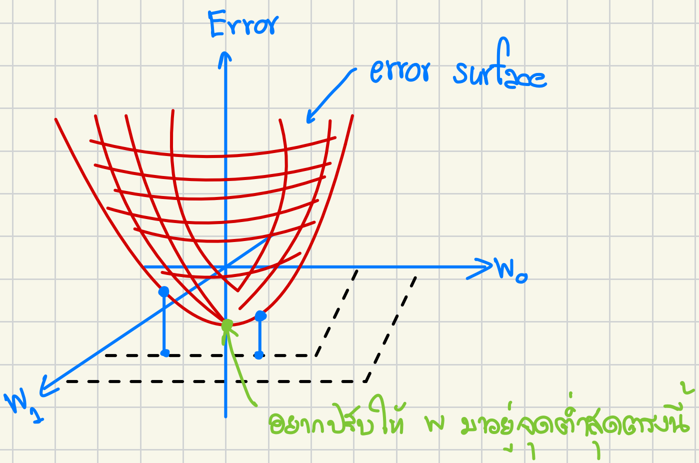


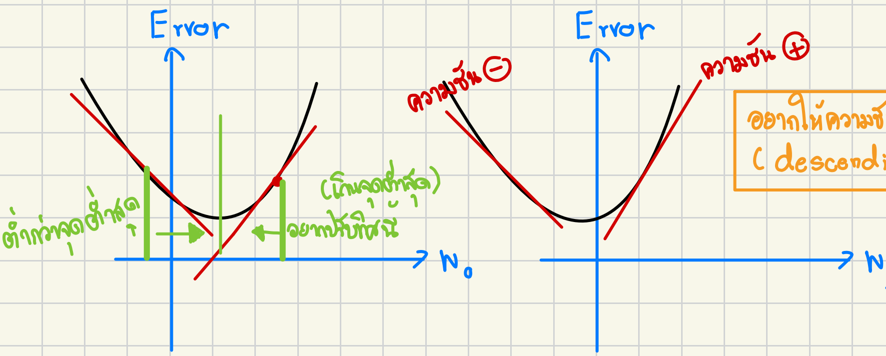
Loss function บอกว่าผิดพลาดขนาดไหน

$\boxed{\text{Error}}$ $e = |target_i - output_i|$

$$e = \frac{1}{2}(target_i - output_i)^2$$

$$output = \boxed{w_0 + w_1 x_1}$$





Gradient Descent

อยากให้ความชัน (Gradient) สุดท้ายลดลง (descending) เป็นศูนย์

2 กราฟนี้พาราโบลาเดียวกัน แต่มอง side view กับ front view ไม่เหมือนกัน

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

$$\vec{w} = \vec{w} - \boxed{\nabla\vec{e}}$$ หาความชัน error เทียบ $\vec{w}$

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

$$= \frac{\partial}{\partial\vec{w}}\frac{1}{2}(\text{target}-\vec{w}\cdot\vec{x})^2 \qquad \frac{d}{dx}u^2 = 2u\frac{du}{dx}$$

$$= \frac{1}{\cancel{2}}\left[\cancel{2}(\text{target}-\vec{w}\cdot\vec{x})\frac{\partial}{\partial\vec{w}}(\cancel{\text{target}}-\vec{w}\cdot\vec{x})\right]$$

ดิฟค่าคงที่ $= 0$

$$= (\text{target}-\vec{w}\cdot\vec{x})(-\vec{x})$$

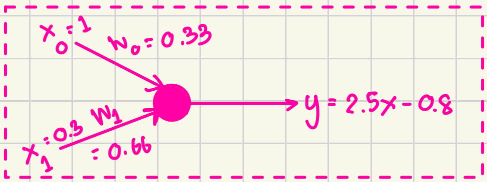
$$= (\text{target}-\text{output})(-\vec{x})$$

$$\boxed{\vec{w} = \vec{w} + (\text{target}-\text{output})(\vec{x})}$$ สมการปรับ weight linear node

---

Training a perceptron example

$y = 2.5x - 0.8$ linear 1 มิติ $\eta = 0.1$



x_1	w_0 rand()	w_1 rand()	output	target	Δw_0	Δw_1
rand(1-10)	w_0	w_1	$w_0 + w_1 x_1$	$-0.8 + 2.5x_1$	$\eta \times (t-o)\cancel{x_0}\,1$	$\eta \times (t-o)x_1$
รอบถัดไป rand()	$= w_0 + \Delta w_0$	$= w_1 + \Delta w_1$	"	"	"	"

Δw_0 สามารถนำไปหารด้วย ขนาด $= \eta \times (t-o)\dfrac{1}{\sqrt{1+x_1^2}}$

Δw_1 สามารถนำไปหารด้วย ขนาด $= \eta \times (t-o)\dfrac{x_1}{\sqrt{1+x_1^2}}$

ความชันของ Sigmoid

$$\boxed{\sigma(y) = \frac{1}{1+e^{-y}}}$$

$y = \sum_i w_i x_i$ หรือ $y = \vec{w} \cdot \vec{x}$

$$\boxed{\frac{\partial}{\partial y}\sigma(y) = \sigma(y)(1-\sigma(y))}$$

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

$$\frac{\partial\, \text{error}}{\partial \vec{w}} = \frac{1}{2}\frac{\partial}{\partial \vec{w}}(t-o)^2$$

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

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

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

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

$$= -(t-o)\underbrace{\sigma(\vec{w}\cdot\vec{x})}_{\text{output}}(1-\underbrace{\sigma(\vec{w}\cdot\vec{x})}_{\text{output}})\vec{x}$$

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

แล้วเอาความชันไปลบ

$$\boxed{\begin{aligned}\vec{w} &= \vec{w} + \Delta\vec{w} \\ \Delta\vec{w} &= \eta(t-o)(o)(1-o)\vec{x}\end{aligned}}$$

สมการปรับ weight sigmoid