

7. Deep Learning :

พื้นฐาน + ความเข้าใจ

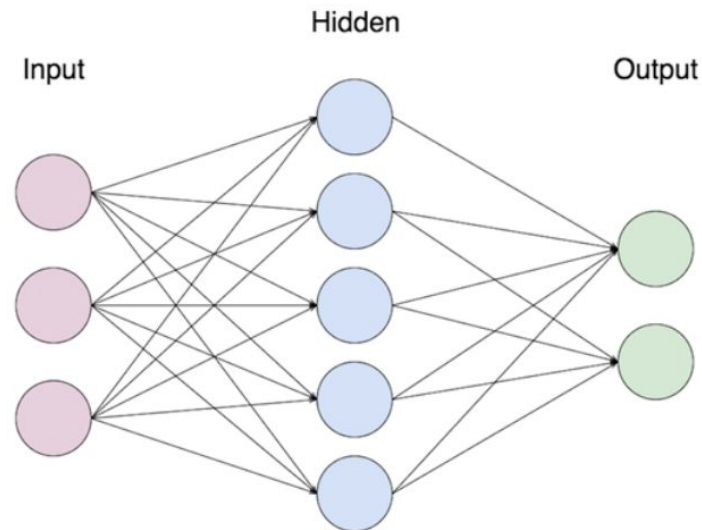
Krittameth Teachasrisaksakul

บทนำ

Deep learning เกี่ยวพันกับการฝึก / สร้าง (training) neural networks (โครงข่ายประสาทเทียม)

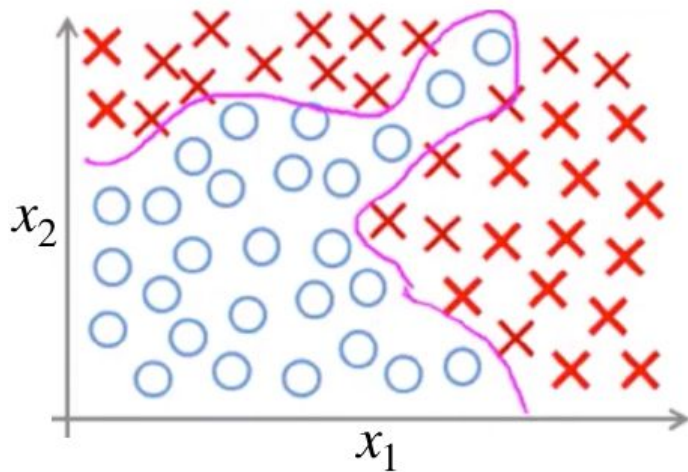
หัวข้อที่จะเรียน

- Neural networks ทำงานอย่างไร
- ทำความเข้าใจ learning algorithms ของ neural network แบบง่าย / พื้นฐาน
- โครงสร้างของ neural network ที่ทันสมัย และมีคนประยุกต์ใช้มาก



ทำไมเราควรใช้ Neural Network ?

เพื่อแก้ปัญหาที่มีลักษณะเป็น non-linearly separable ก็คือ แยกชนิดข้อมูลได้ด้วย decision boundary ที่ไม่เป็นเชิงเส้น



การแก้ปัญหาแบบ non-linearly separable

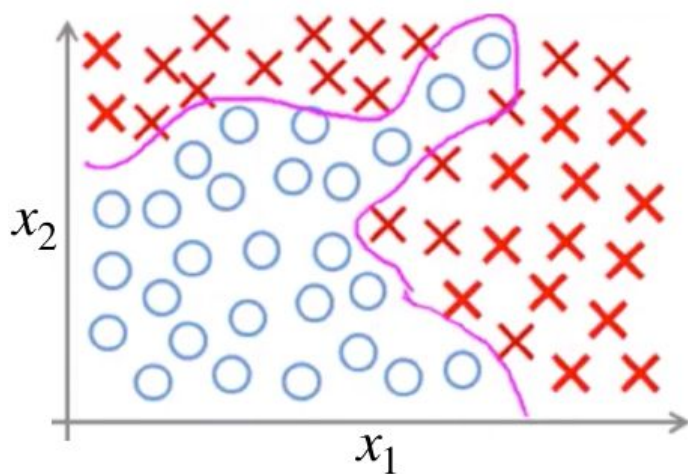
Non-linear

classification

:

การแยกประเภทด้วย decision boundary ที่ไม่เป็นเชิงเส้น

$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 x_2 + \theta_5 x_1^3 x_2 + \theta_6 x_1 x_2^2 + \dots)$$



วิธีนี้เป็นไปได้ แต่มันเป็นวิธีแก้ที่ตีหรือเปล่า ?

ถ้าไม่ ทำไมมันจึงไม่ใช่วิธีแก้ที่ดี ?

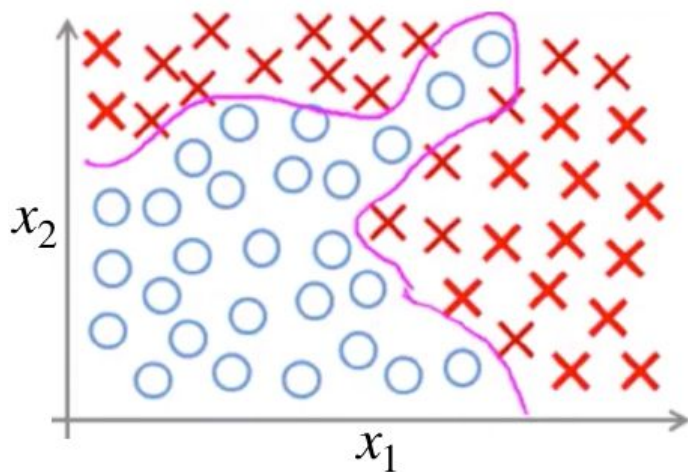
การแก้ปัญหาแบบ non-linearly separable

Non-linear

classification

:

การแยกประเภทด้วย decision boundary ที่ไม่เป็นเชิงเส้น



$x_1 = \text{size}$
 $x_2 = \text{\#bedrooms}$
 $x_3 = \text{\#floor}$
 $x_4 = \text{age}$
 \vdots
 x_{100}

} 100 features

ถ้าจะสร้าง model ที่เป็นพหุนาม (polynomial) ที่มี degree เป็น 2 จะต้องใช้พจน์กี่พจน์ในสมการ ?

$x_1^2, x_1x_2, x_1x_3, x_1x_4, \dots, x_1x_{100}$
 $x_2^2, x_2x_3, x_2x_4, \dots, x_2x_{100}$
 \vdots

} ≈ 5000 terms

This number grows in $\mathcal{O}(n^2)$

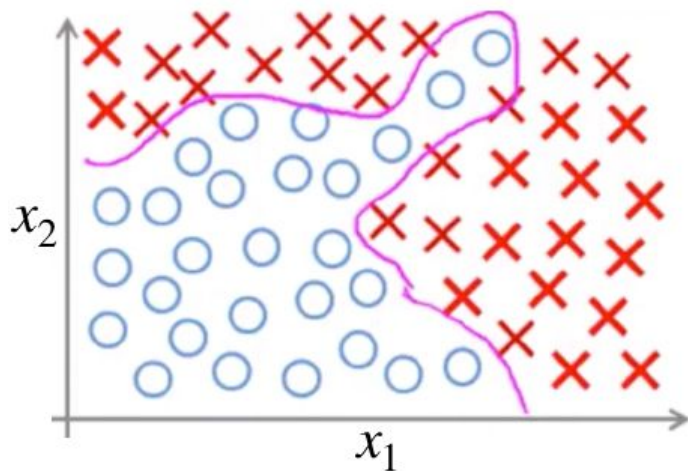
การแก้ปัญหาแบบ non-linearly separable

Non-linear

classification

:

การแยกประเภทด้วย decision boundary ที่ไม่เป็นเชิงเส้น



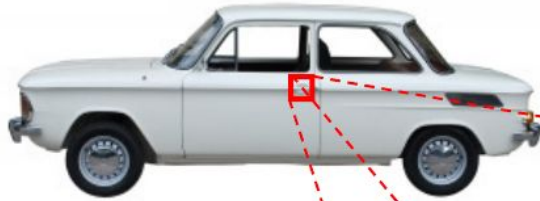
$$\left. \begin{array}{l} x_1 = \text{size} \\ x_2 = \text{\#bedrooms} \\ x_3 = \text{\#floor} \\ x_4 = \text{age} \\ \vdots \\ x_{100} \end{array} \right\} 100 \text{ features}$$

ถ้าจะสร้าง model ที่เป็นพหุนาม (polynomial) ที่มี degree เป็น 2 จะต้องใช้พจน์กี่พจน์ในสมการ ?

$$\begin{aligned} \text{ถ้าเป็น } & x_1 x_2 x_3, x_1^2 x_3, x_{10} x_{11} x_{17}, \dots ? \\ & \approx 170,000 \text{ features i.e. } \mathcal{O}(n^3) \end{aligned}$$

ปัญหา Computer Vision (คอมพิวเตอร์วิทัศน์)

คนเห็นรูปภาพนี้



brightness (ความสว่าง) ของ pixel ในรูปภาพ (หรือ intensity value : ค่าความเข้มของสี)

กล้องเห็นภาพนี้เป็น

194	210	201	212	199	213	215	195	178	158	182	209
180	189	190	221	209	205	191	167	147	115	129	163
114	126	140	188	176	165	152	140	170	106	78	88
87	103	115	154	143	142	149	153	173	101	57	57
102	112	106	131	122	138	152	147	128	84	58	66
94	95	79	104	105	124	129	113	107	87	69	67
68	71	69	98	89	92	98	95	89	88	76	67
41	56	68	99	63	45	60	82	58	76	75	65
20	43	69	75	56	41	51	73	55	70	63	44
50	50	57	69	75	75	73	74	53	68	59	37
72	59	53	66	84	92	84	74	57	72	63	42
67	61	58	65	75	78	76	73	59	75	69	50

คำถาม

สมมติ เราเรียนรู้การจดจำรถยนต์จากภาพขนาด 100×100 pixel ให้ features เป็น intensity value (ค่าความเข้มของสี) ของ pixel ถ้าเราสร้างโมเดล logistic regression ที่มีพจน์กำลังสอง (quadratic terms) $(\mathbf{X}_i \mathbf{X}_j)$ เป็น features เราจะมี features กี่ตัว ?

(i) 5,000

(ii) 100,000

(iii) 50 ล้าน (5×10^7)

(iv) 5 พันล้าน (5×10^9)

คำใบ้:
$$C(n, r) = \frac{n!}{r!(n - r!)}$$

คำถาม

สมมติ เราเรียนรู้การจดจำรถยนต์จากภาพขนาด 100×100 pixel ให้ features เป็น intensity value (ค่าความเข้มของสี) ของ pixel ถ้าเราสร้างโมเดล logistic regression ที่มีพจน์กำลังสอง (quadratic terms) $(\mathbf{X}_i \mathbf{X}_j)$ เป็น features เราจะมี features กี่ตัว ?

(i) 5,000

(ii) 100,000

(iii) 50 ล้าน (5×10^7)

(iv) 5 พันล้าน (5×10^9)

คำใบ้:
$$C(n, r) = \frac{n!}{r!(n-r)!}$$

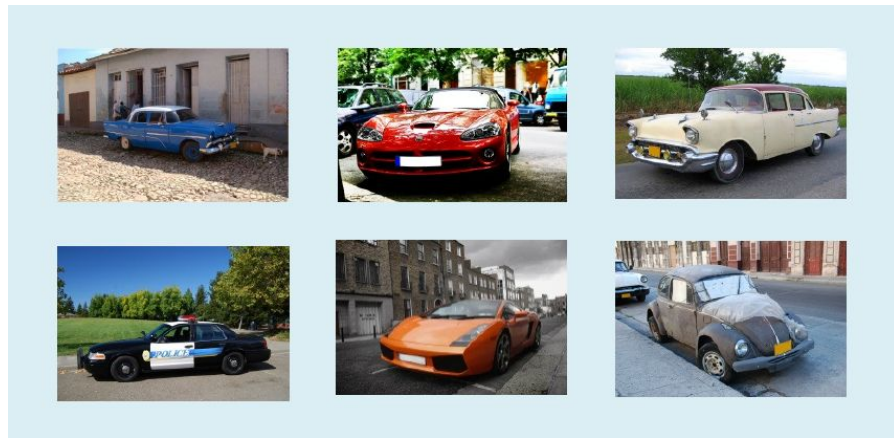
วิธีทำ:

รูปภาพ 100×100 pixel $\Rightarrow 10,000$ pixels ($\therefore n = 10000$)

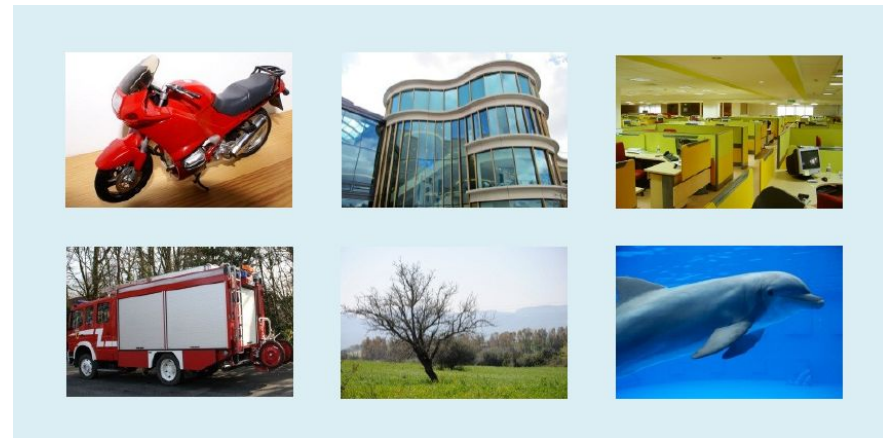
Quadratic features

$$\begin{aligned} C(10000, 2) &= 10000! / (2! (10000-2)!) \\ &= 10000! / (2! 9998!) \\ &= 10000 \times 9999 / 2 \\ &\approx 5 \times 10^7 \text{ features} \end{aligned}$$

Car Detection : การตรวจหารถยนต์ (จากภาพ)



ตัวอย่างภาพ “รถยนต์” (cars)

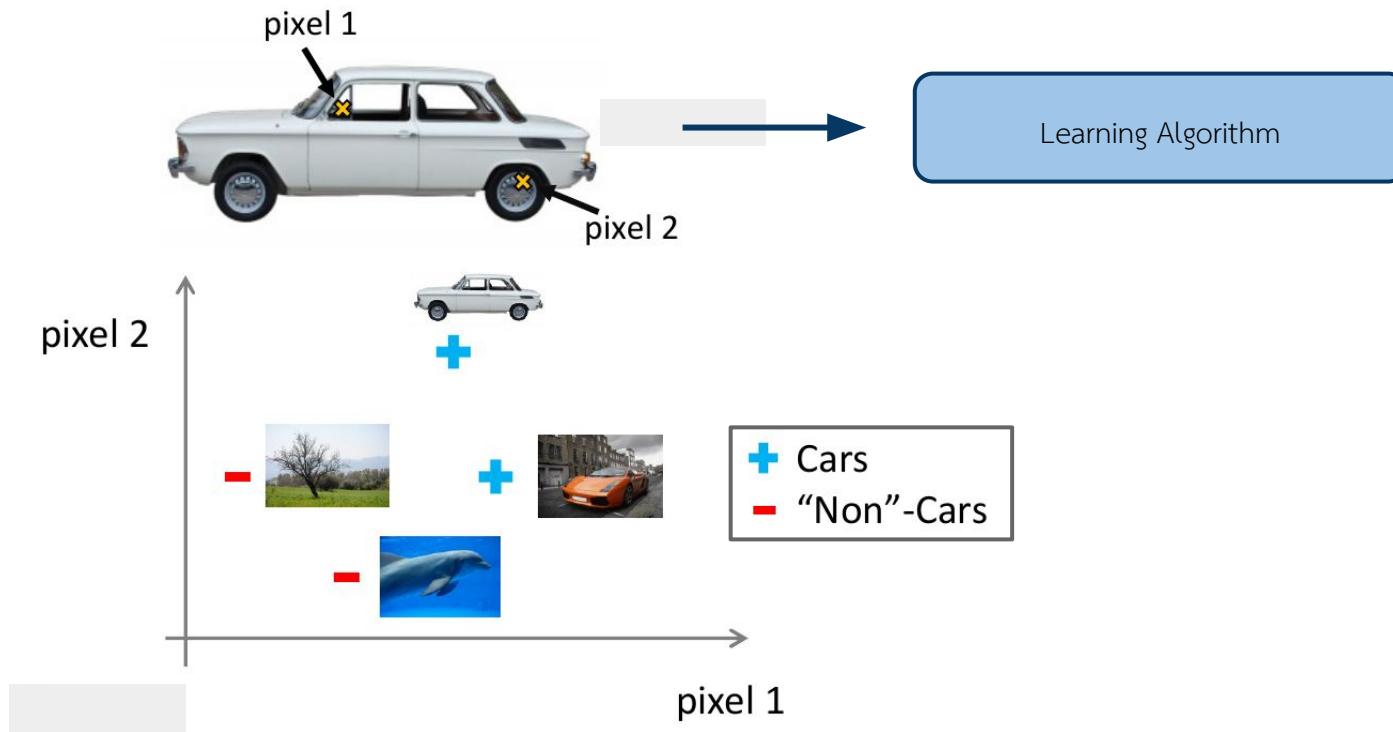


ตัวอย่างภาพ “ไม่ใช่รถยนต์” (not cars)

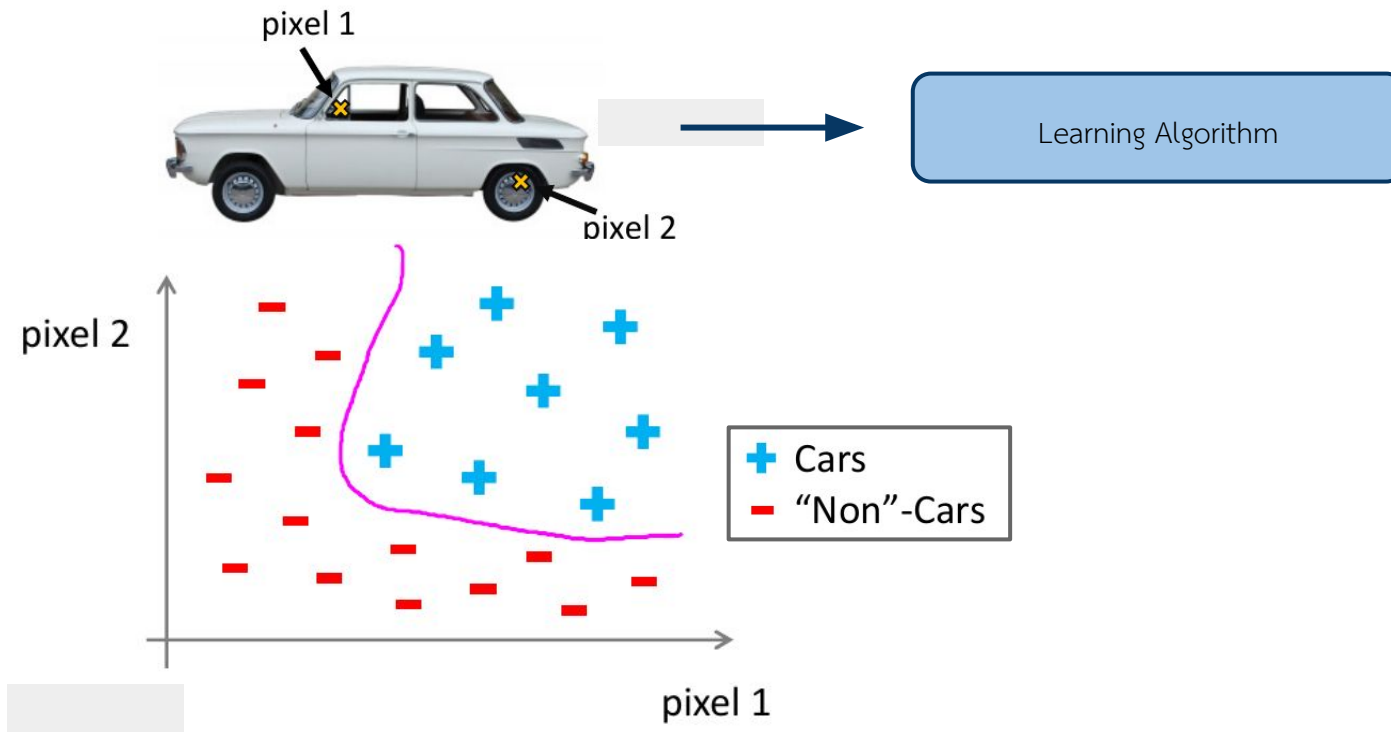
Testing:
ภาพนี้อยู่ในกลุ่ม/ชนิดไหน?



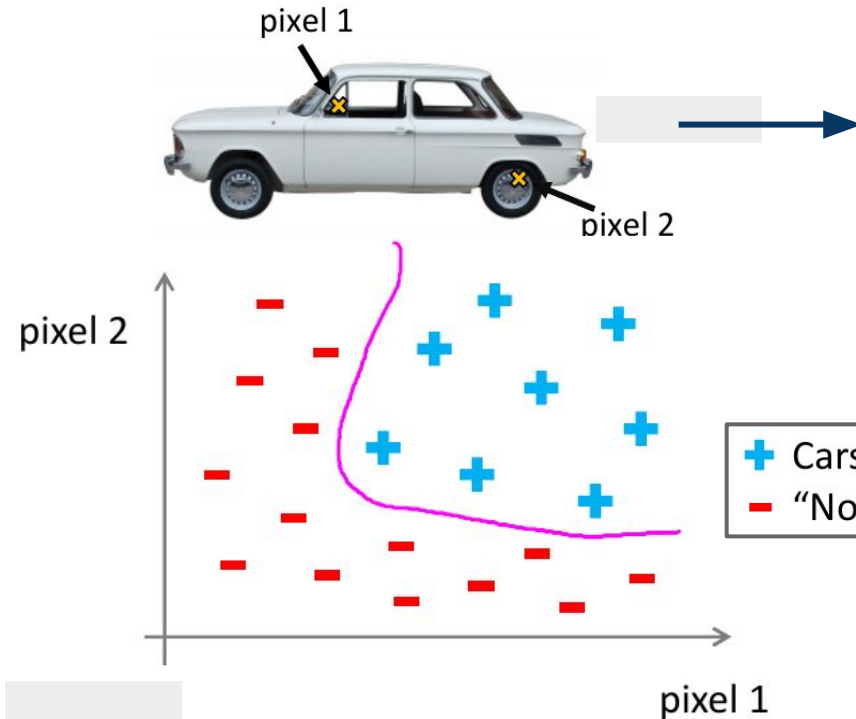
Car Detection



Car Detection



Car Detection



Quadratic features

$$\begin{aligned} C(2500, 2) &= 2500! / (2! (2500-2)!) \\ &= 2500! / (2! 2498!) \\ &= 2500 \times 2499 / 2 \\ &\approx 3 \times 10^6 \text{ features} \end{aligned}$$

Q: What is the dimension of the feature space?

A: รูปภาพ 50×50 pixel
 $\Rightarrow 2,500$ pixels ($\therefore n = 2500$)

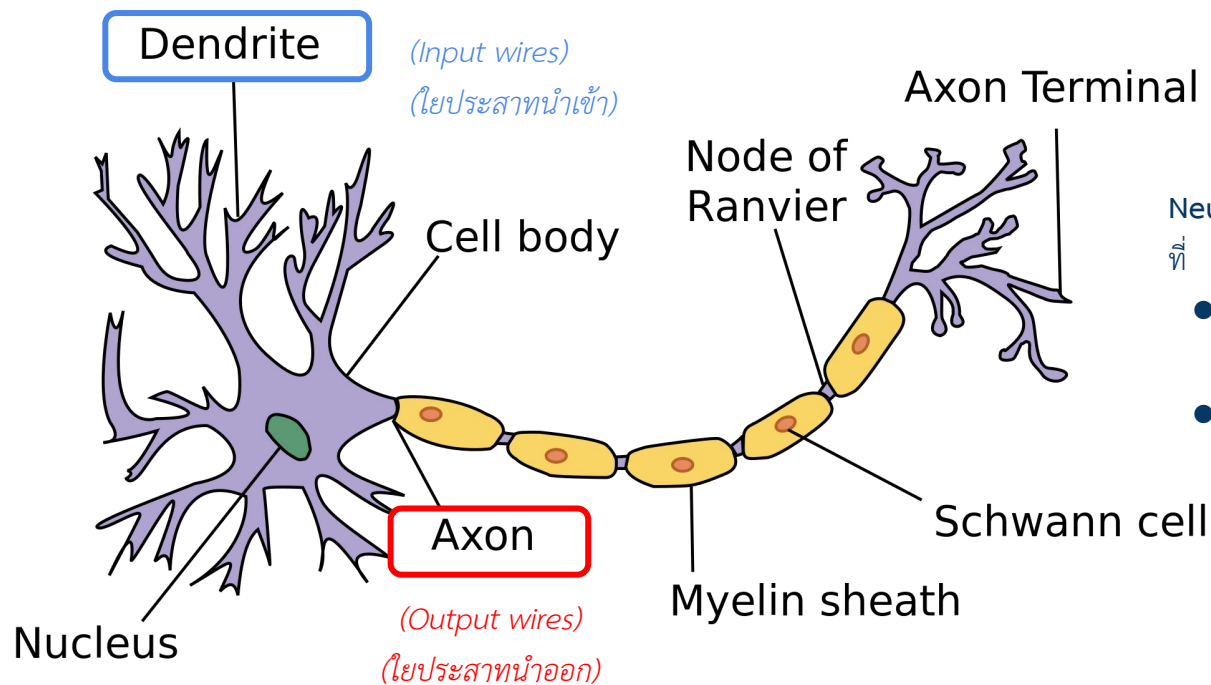
Quadratic features (i.e $X_i \times X_j$)
 $\approx 3 \times 10^6$ features

7.1 พื้นฐาน Neural Network

เขียนแทน hypothesis function ด้วย neural networks ได้อย่างไร

Krittameth Teachasrisaksakul

Neuron (เซลล์ประสาท) ในสมอง



Neuron เป็นหน่วยคำนวณ (computational unit) ที่

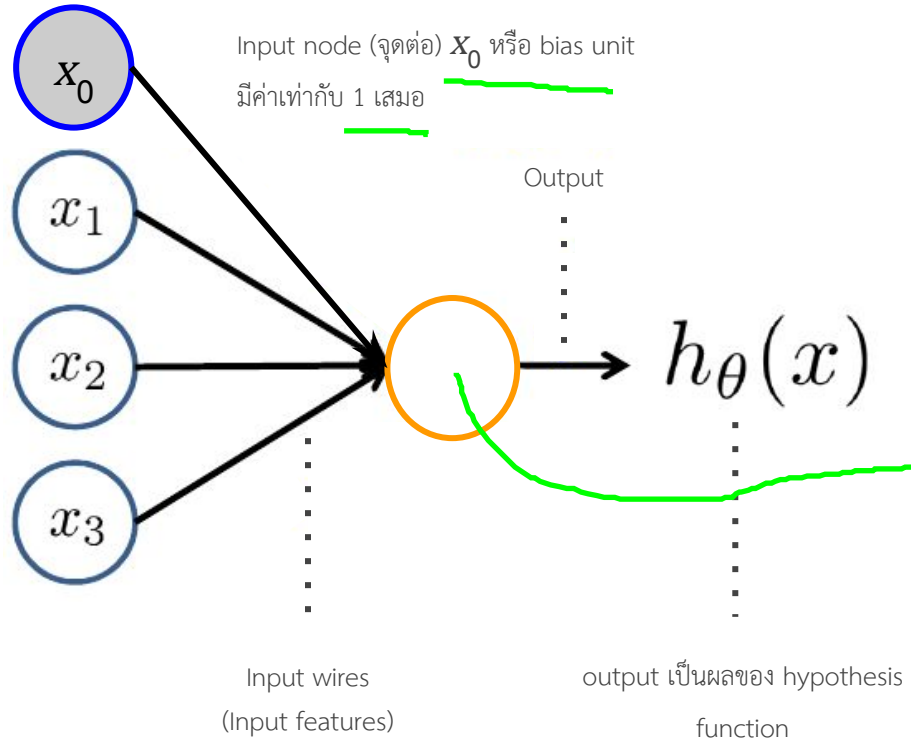
- รับ input เป็นสัญญาณไฟฟ้า (ผ่าน dendrites)
- และส่งออกเป็น output (ผ่าน axons)

Figure source: "Anatomy and Physiology" by the US National Cancer Institute's Surveillance,

Epidemiology and End Results (SEER) Program .

<https://commons.wikimedia.org/wiki/File:Neuron.svg>

Neuron Model (แบบจำลอง neuron): Logistic Unit



$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

Input features

weights

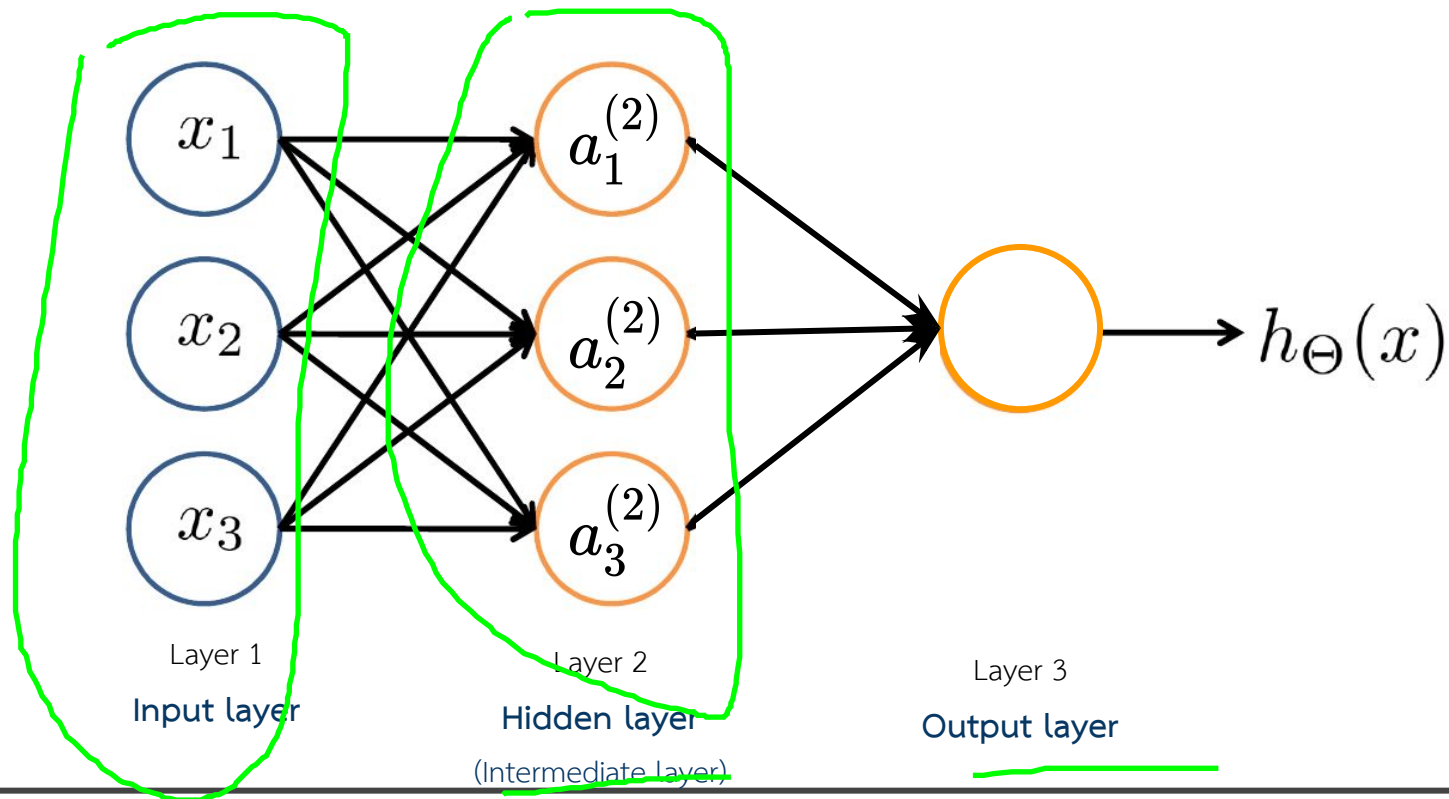
Activation

ใช้ sigmoid (logistic) function ซึ่งเคยใช้ทำ classification

function:

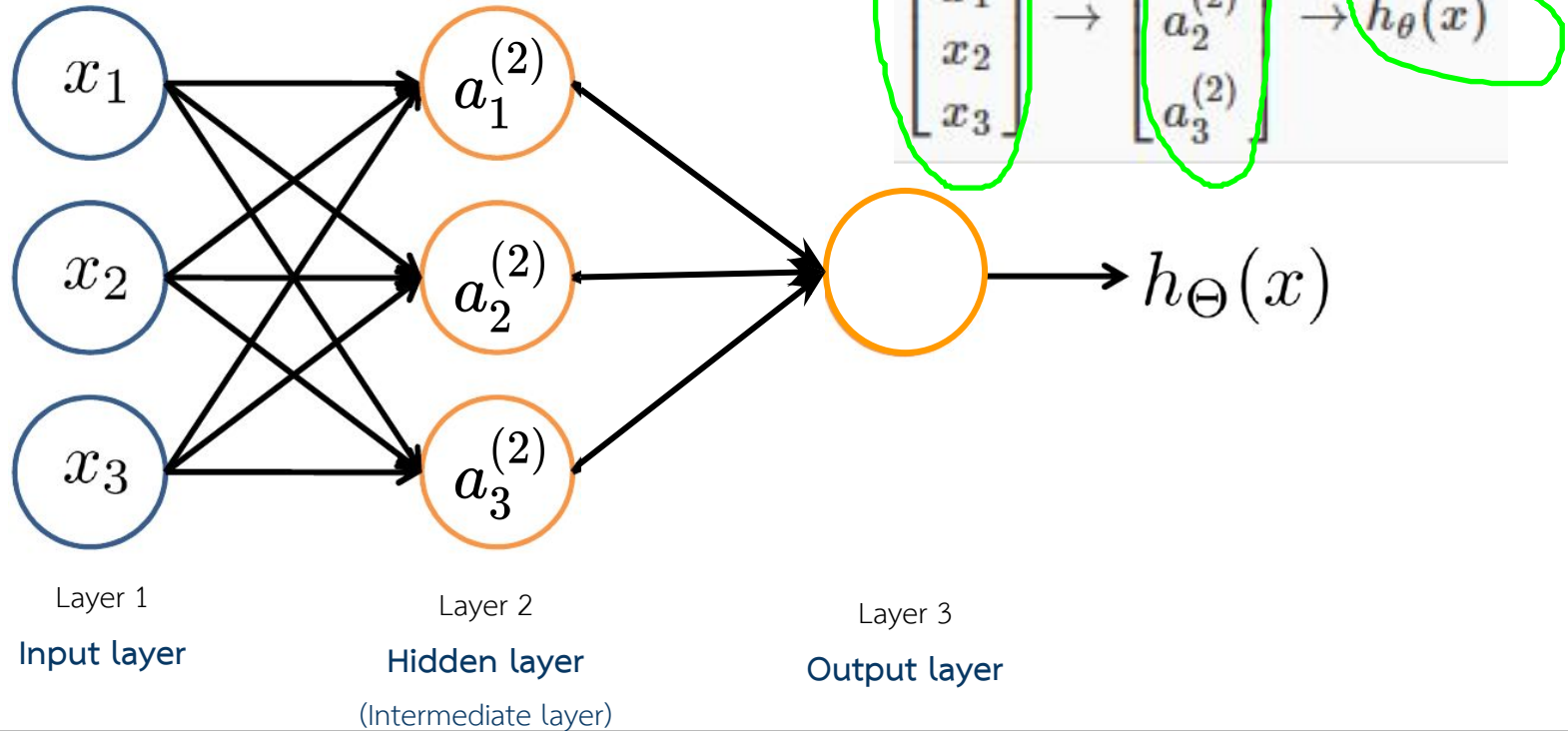
$$g(z) = \frac{1}{1+e^{-z}}$$

Neural Network

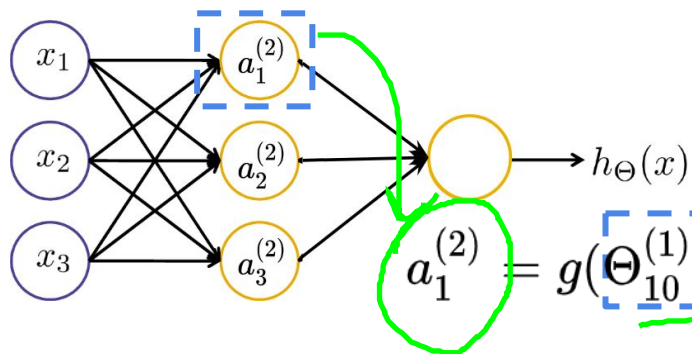


เขียนแทน แบบง่ายๆ ด้วย

Neural Network



Neural Network



$a_i^{(j)}$ = "activation" ของ unit (หน่วย) i ใน layer (ชั้น) j

$\Theta^{(j)}$ = matrix ของ weights ที่ควบคุม function mapping from layer j to layer $j+1$

$$a_1^{(2)} = g(\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3)$$

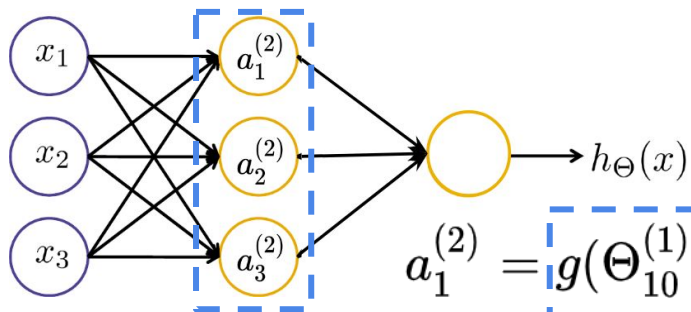
หาค่า activation unit 1 unit ใน hidden layer โดย

1. คูณ 1 แถวของ matrix parameters $\Theta^{(1)}$ กับ input vector X

2. ใช้ logistic function $g(\cdot)$ กับ ผลจาก ข้อ 1

$$\begin{bmatrix} \Theta_{10}^{(1)} & \Theta_{11}^{(1)} & \Theta_{12}^{(1)} & \Theta_{13}^{(1)} \\ \Theta_{20}^{(1)} & \Theta_{21}^{(1)} & \Theta_{22}^{(1)} & \Theta_{23}^{(1)} \\ \Theta_{30}^{(1)} & \Theta_{31}^{(1)} & \Theta_{32}^{(1)} & \Theta_{33}^{(1)} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} = \Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3$$

Neural Network



$a_i^{(j)}$ = "activation" ของ unit (หน่วย) i ใน layer (ชั้น) j

$\Theta^{(j)}$ = matrix ของ weights ที่ควบคุม function mapping from layer j to layer $j+1$

$$\begin{aligned} a_1^{(2)} &= g(\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3) \\ a_2^{(2)} &= g(\Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{23}^{(1)} x_3) \\ a_3^{(2)} &= g(\Theta_{30}^{(1)} x_0 + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3) \end{aligned}$$

$$\Theta^{(1)} \in \mathbb{R}^{3 \times 4}$$

• แต่ละ layer (ชั้น) จะมี matrix ของ weights $\Theta^{(j)}$ ของมันเอง

• ถ้าใน network

- layer j มี s_j unit (หน่วย) และ
- layer $j+1$ มี s_{j+1} unit
- แล้ว $\Theta^{(j)}$ จะมี dimension (มิติ) เป็น $s_{j+1} \times (s_j + 1)$

output nodes ไม่รวม bias nodes

$s_j + 1$

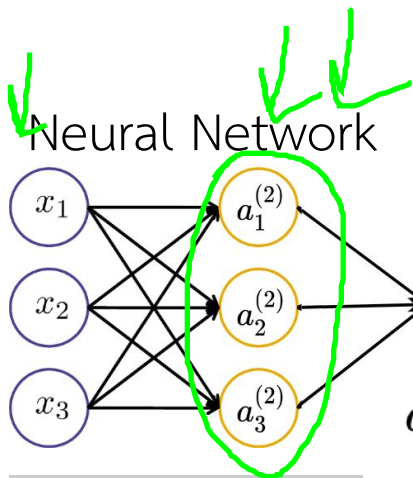
layer 1 มี 3 unit, layer 2 มี 3 unit

—> dimension ของ $\Theta^{(j)}$

$$= s_{j+1} \times (s_j + 1) = 3 \times (3+1) = 3 \times 4$$

input nodes นับรวม bias

node x_0 ด้วย



hypothesis output

$a_i^{(j)}$ = "activation" ของ unit (หน่วย) i ใน layer (ชั้น) j

$\Theta^{(j)}$ = matrix ของ weights ที่ควบคุม function mapping from layer j to layer $j+1$

layer 2 มี 3 unit, layer 3 มี 1 unit

dimension ของ $\Theta^{(j)}$

$= s_{j+1} \times (s_j + 1) = 1 \times (3+1) = 1 \times 4$

$\Theta^{(2)} \in \mathbb{R}^{1 \times 4}$

$$a_1^{(2)} = g(\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3)$$

$$a_2^{(2)} = g(\Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{23}^{(1)} x_3)$$

$$a_3^{(2)} = g(\Theta_{30}^{(1)} x_0 + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3)$$

$$h_{\Theta}^{(x)} = a_1^{(3)} = g(\Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)})$$

$$\begin{bmatrix} \Theta_{10}^{(2)} & \Theta_{11}^{(2)} & \Theta_{12}^{(2)} & \Theta_{13}^{(2)} \end{bmatrix}$$

Weight matrix ของ layer 2

$$\begin{bmatrix} a_0^{(2)} \\ a_1^{(2)} \\ a_2^{(2)} \\ a_3^{(2)} \end{bmatrix}$$

ค่าจาก activation nodes

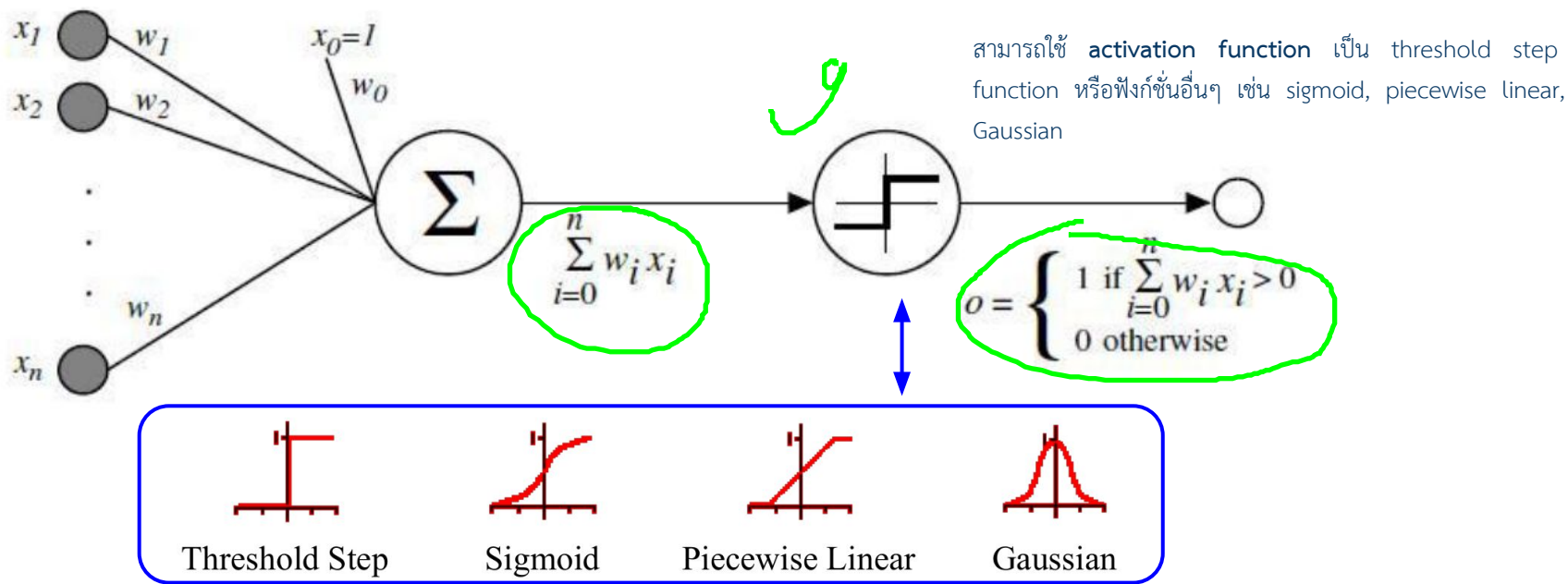
7.2 Neural Network:

ตัวอย่างและความเข้าใจพื้นฐาน

Krittameth Teachasrisaksakul

ตัวอย่างและความเข้าใจพื้นฐาน

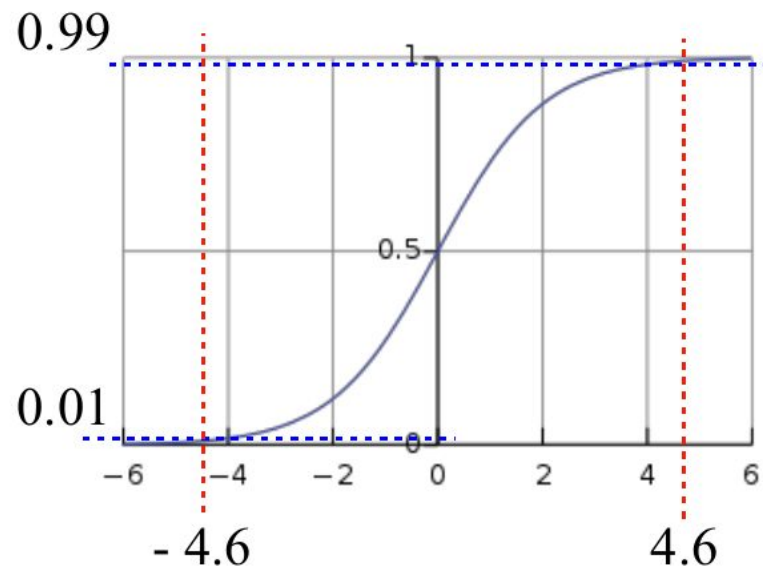
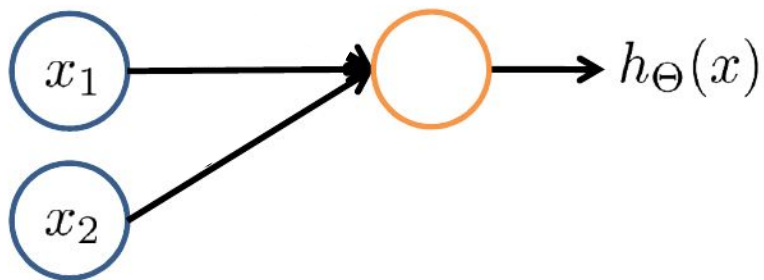
Neural network model หลายๆอัน ที่ใช้ทุกวันนี้ มีรากฐานมาจาก Perceptron ของ Rosenblatt (1958)



ตัวอย่าง: Logical AND operator

$$x_1, x_2 \in \{0, 1\}$$

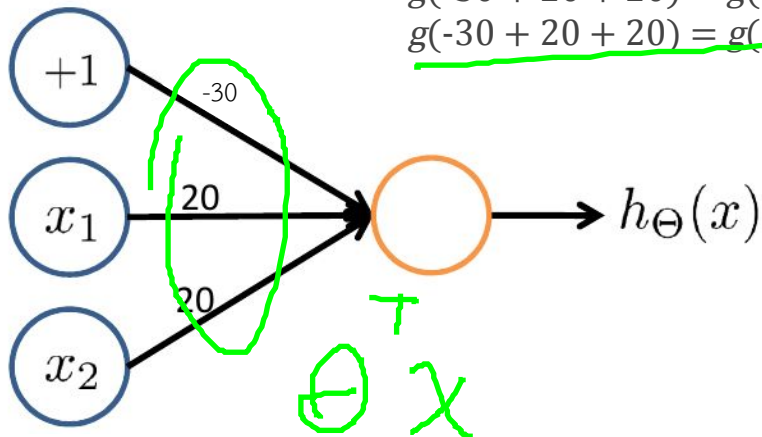
$$y = x_1 \text{ AND } x_2$$



ตัวอย่าง: Logical AND operator

$$x_1, x_2 \in \{0, 1\}$$

$$y = x_1 \text{ AND } x_2$$



$$h_{\Theta}(x) = g(-30 + 20x_1 + 20x_2)$$

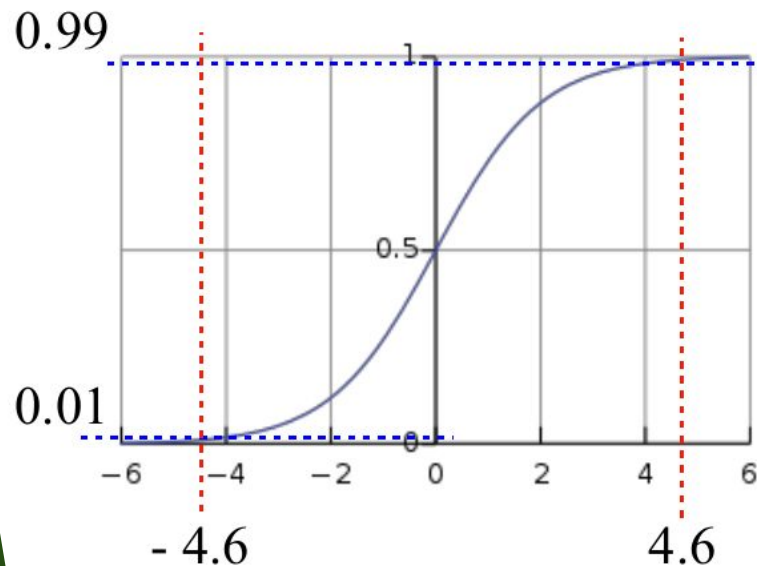
วิธีทำ:

$$g(-30 + 0 + 0) = g(-30) \approx 0$$

$$g(-30 + 0 + 20) = g(-10) \approx 0$$

$$g(-30 + 20 + 20) = g(10) \approx 1$$

$$g(-30 + 20 + 20) = g(10) \approx 1$$

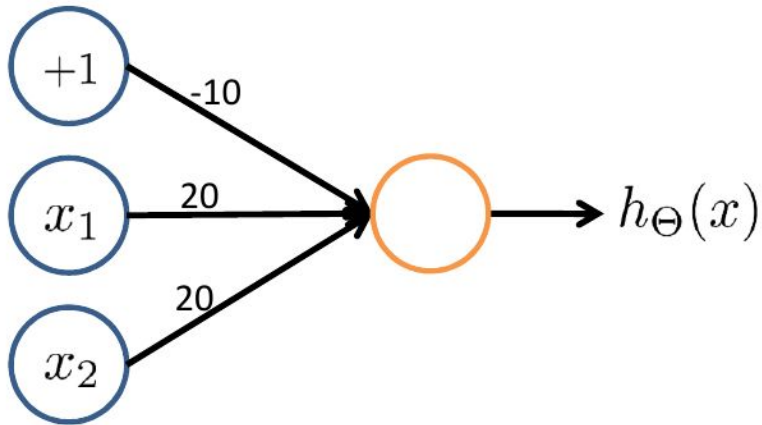


x_1	x_2	$h_{\Theta}(x)$
0	0	$g(-30) \approx 0$
0	1	$g(-10) \approx 0$
1	0	$g(-10) \approx 0$
1	1	$g(10) \approx 1$

Question

สมมติ X_1 และ X_2 เป็นเลขฐานสอง (มีค่าเป็น 0 หรือ 1) network ด้านล่างจะคำนวณ boolean function อะไร ?

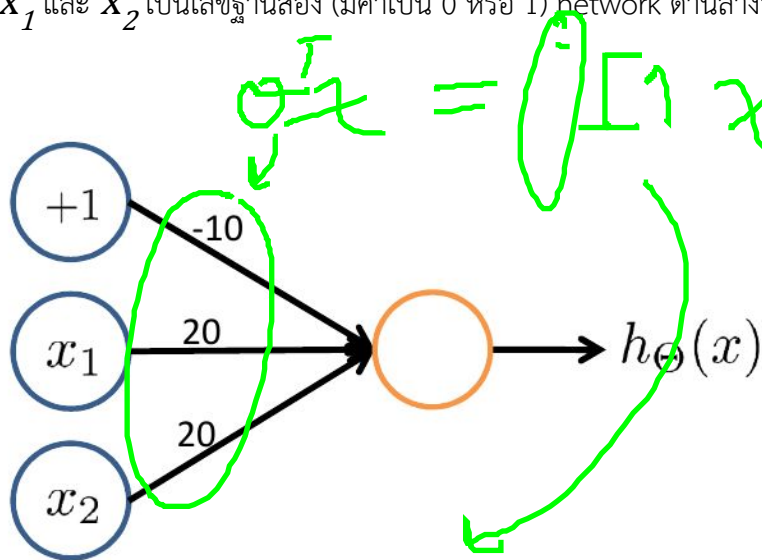
0



Question

$$0 \vee 0 = 0$$

สมมติ x_1 และ x_2 เป็นเลขฐานสอง (มีค่าเป็น 0 หรือ 1) network ด้านล่างจะคำนวณ boolean function อะไร ?



$$h_{\Theta}(x) = g(-10 + 20x_1 + 20x_2)$$

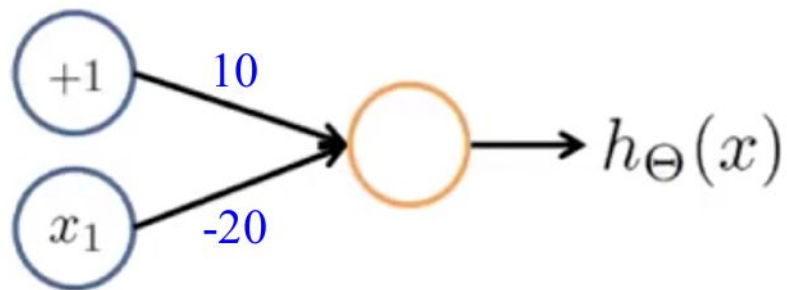
ตอบ: Logical OR

x_1	x_2	$h_{\Theta}(x)$
0	0	$g(-10) \approx 0$
0	1	$g(10) \approx 1$
1	0	$g(10) \approx 1$
1	1	$g(30) \approx 1$

วิธีทำ:

$$\begin{aligned} g(-10 + 0 + 0) &= g(-10) \approx 0 \\ g(-10 + 0 + 20) &= g(10) \approx 1 \\ g(-10 + 20 + 0) &= g(10) \approx 1 \\ g(-10 + 20 + 20) &= g(30) \approx 1 \end{aligned}$$

ตัวอย่าง: Logical NEGATION (นิเสธ)



$$h_{\Theta}(x) = g(+10 - 20x_1)$$

$$\sim 0 = 1$$

x_1	$h_{\Theta}(x)$
0	$g(+10) \approx 1$
1	$g(-10) \approx 0$

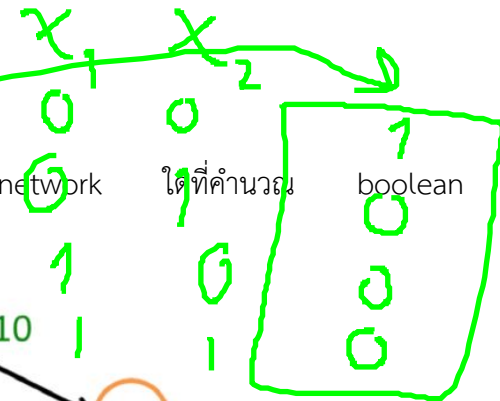
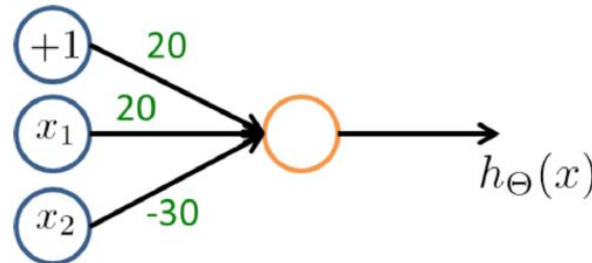
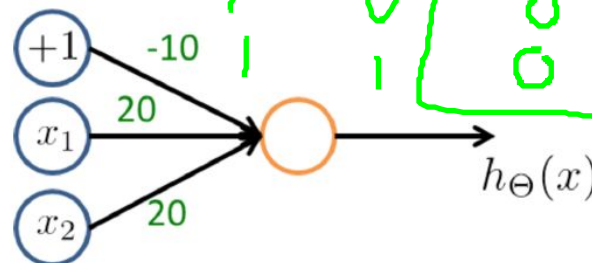
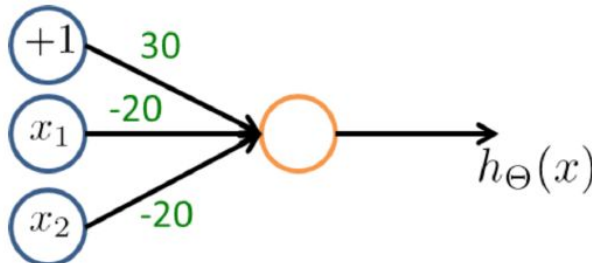
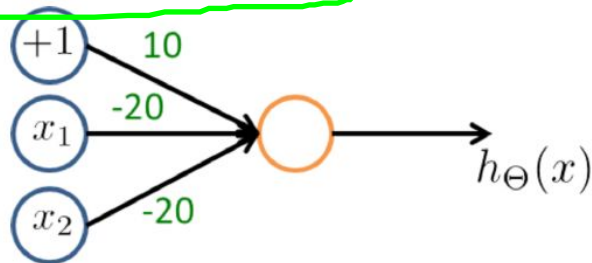
วิธีทำ:

$$\begin{aligned} g(+10 - 0) &= g(+10) \approx 1 \\ g(+10 - 20) &= g(-10) \approx 0 \end{aligned}$$

Question

สมมติ x_1 และ x_2 เป็นเลขฐานสอง (มีค่าเป็น 0 หรือ 1) network ได้ที่คำนวณ boolean function

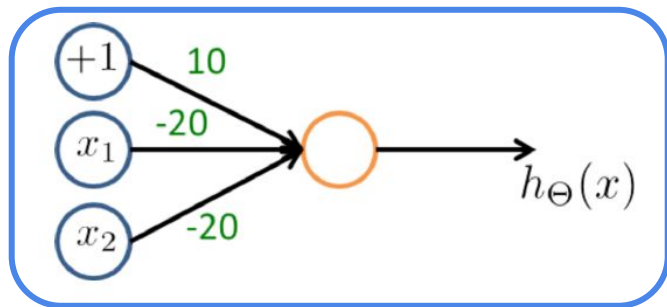
(NOT x_1) AND (NOT x_2)



Question & Answer

สมมติ x_1 และ x_2 เป็นเลขฐานสอง (มีค่าเป็น 0 หรือ 1) network ใดที่คำนวณ boolean function **(NOT x_1) AND (NOT x_2)**

ตอบ:



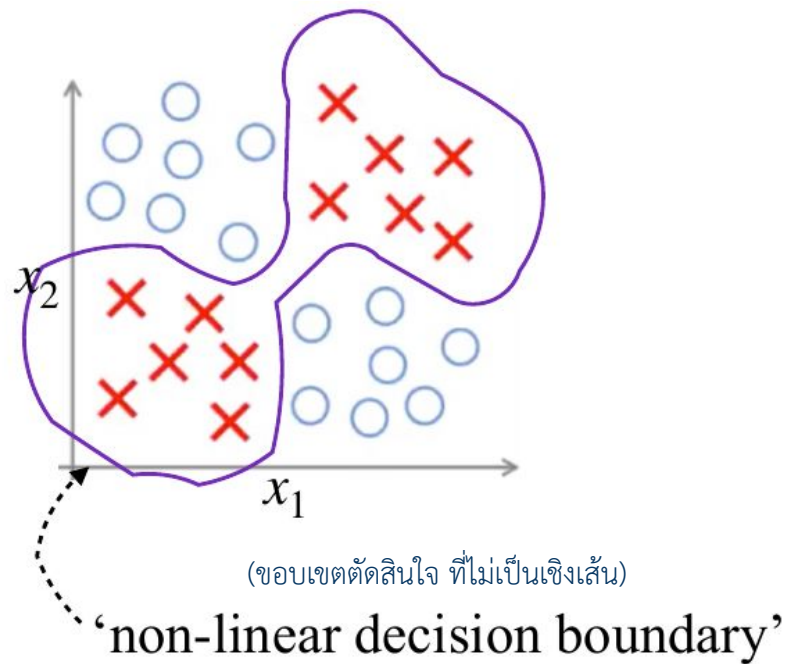
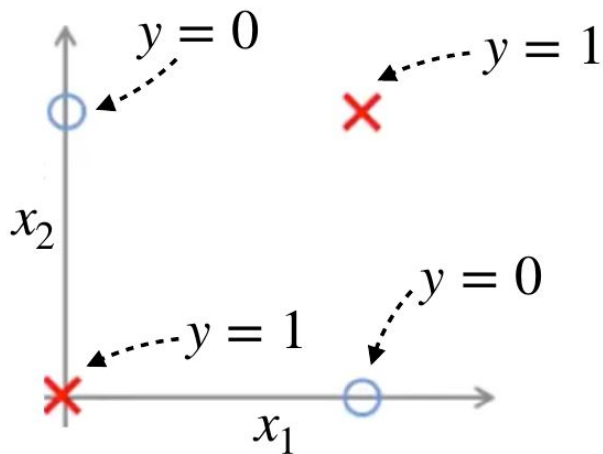
$$(\text{NOT } x_1) \text{ AND } (\text{NOT } x_2) = \text{NOT } (x_1 \text{ OR } x_2)$$

- $= 1$ ก็ต่อเมื่อ $x_1 = 0$ และ $x_2 = 0$

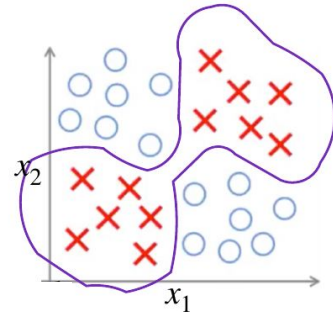
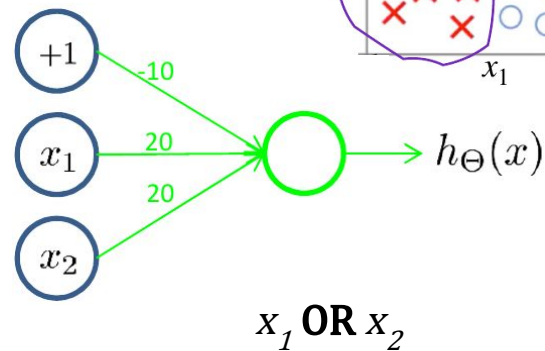
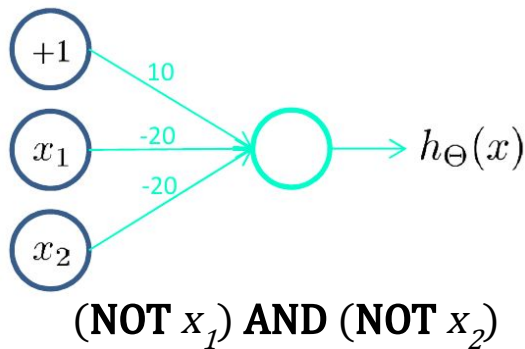
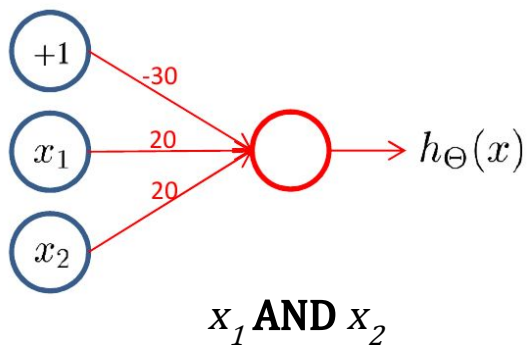
x_1	x_2	$10 - 20x_1 - 20x_2$	$h_{\theta}(x)$
0	0	10	0
0	1	-10	1
1	0	-10	1
1	1	-30	1

ตัวอย่าง: Logical XNOR

X_1 และ X_2 เป็นเลขฐานสอง (มีค่าเป็น 0 หรือ 1)

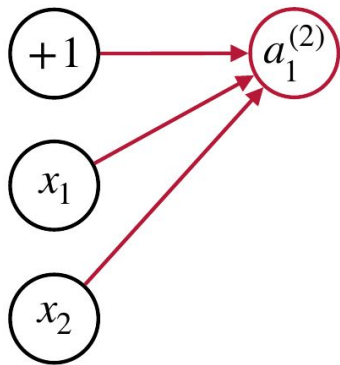
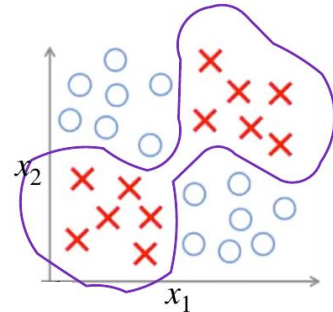
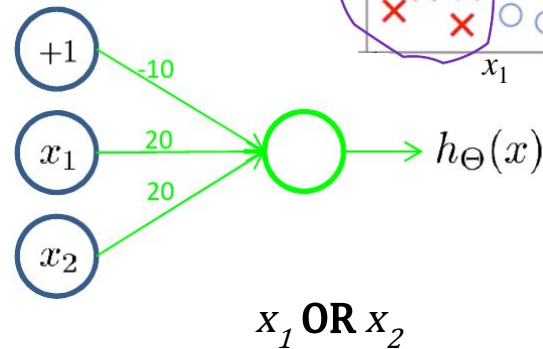
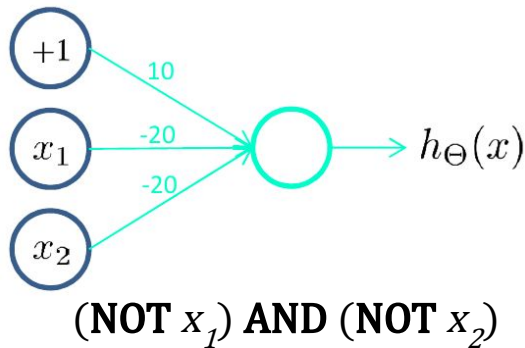
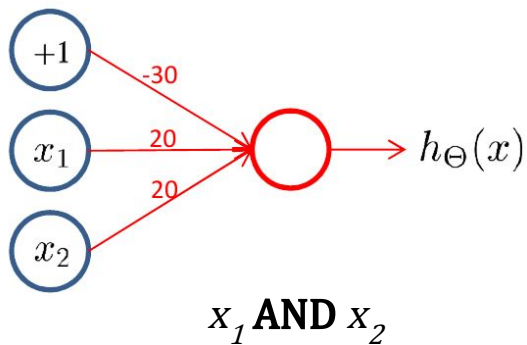


ตัวอย่าง: Logical XNOR



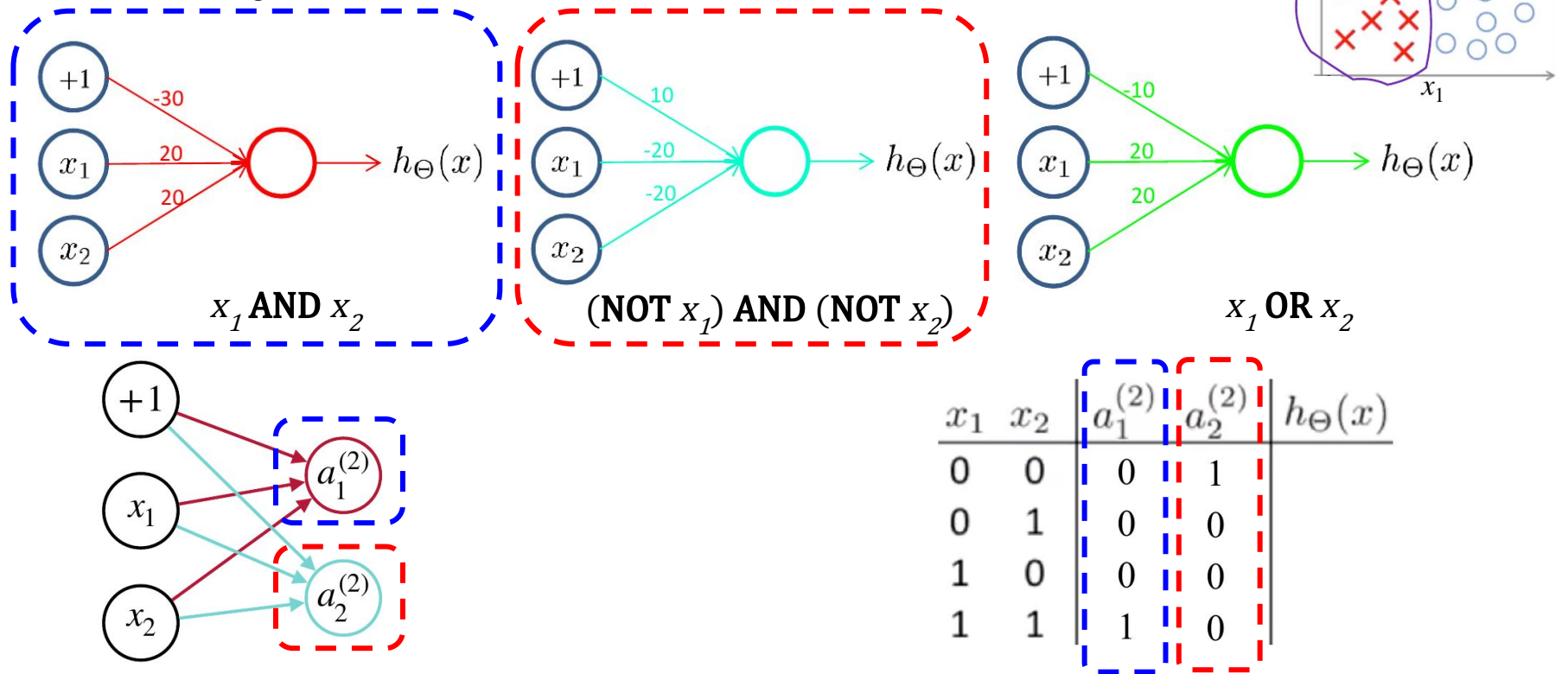
x_1	x_2	$a_1^{(2)}$	$a_2^{(2)}$	$h_{\Theta}(x)$
0	0			
0	1			
1	0			
1	1			

ตัวอย่าง: Logical XNOR



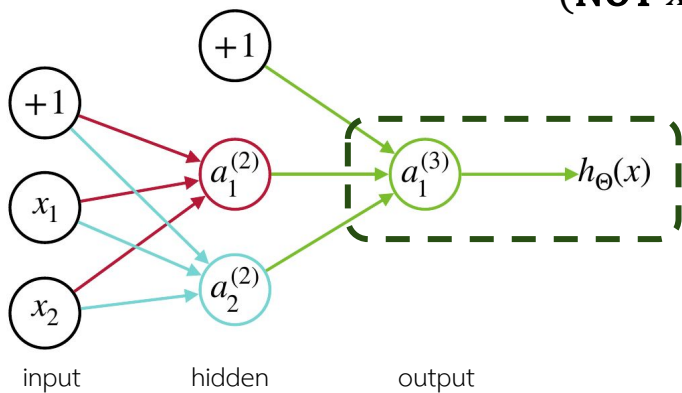
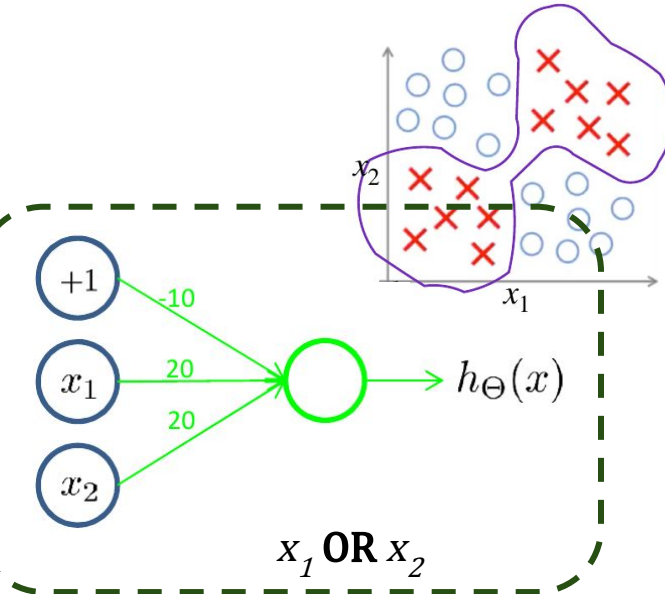
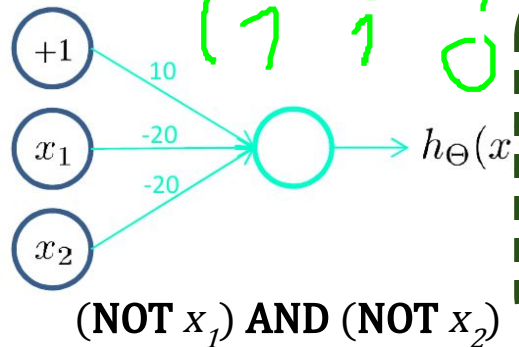
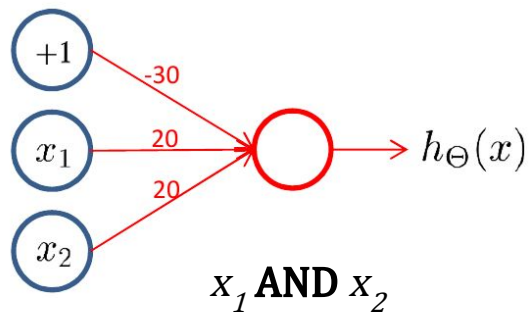
x_1	x_2	$a_1^{(2)}$	$a_2^{(2)}$	$h_{\Theta}(x)$
0	0			
0	1			
1	0			
1	1			

ตัวอย่าง: Logical XNOR



ตัวอย่าง: Logical XNOR

XOR { 0 1 1
1 0 1 }



x_1	x_2	$a_1^{(2)}$	$a_2^{(2)}$	$h_{\Theta}(x)$
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1

7.3 Neural Network: Multi-class Classification

(การแยกประเภท มากกว่า 2 ประเภท)

Krittameth Teachasrisaksakul

k -classes Classification

เพื่อจำแนกข้อมูลเป็น class หลาย class



Pedestrian



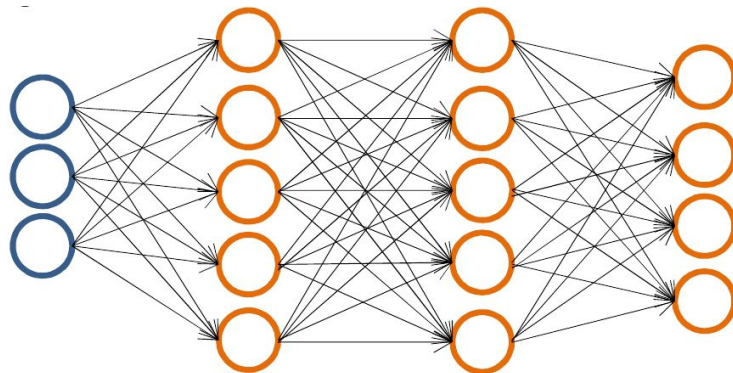
Car



Motorcycle



Truck



(1) algorithm นี้ รับภาพเป็น input และจำแนกประเภทของรูปภาพ

$$h_{\Theta}(x) \in \mathbb{R}^4$$

(2) ตัวอย่าง: อยากแยกข้อมูล เป็นหนึ่งใน 4 ประเภท / category / class \rightarrow ต้องให้ hypothesis function คืนค่า vector ของค่า



$$h_{\Theta}(x) \approx \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



$$h_{\Theta}(x) \approx \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$



$$h_{\Theta}(x) \approx \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$



$$h_{\Theta}(x) \approx \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

k -classes Classification

Goal (เป้าหมาย):



$$h_{\Theta}(x) \approx$$

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



$$h_{\Theta}(x) \approx$$

$$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$



$$h_{\Theta}(x) \approx$$

$$h_{\Theta}(x)_3$$

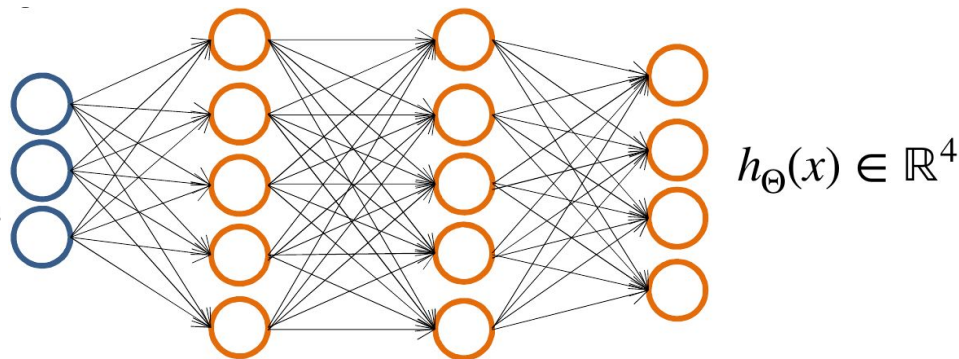
$$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$



$$h_{\Theta}(x) \approx$$

$$h_{\Theta}(x)_4$$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$



Training set (ชุดข้อมูลที่ใช้ฝึก / สร้าง model):

$$(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})$$

เมื่อ $y^{(i)}$ มีค่าเป็น 1 ใน 4 ค่า ดังนี้

class ที่ได้จากการแยกประเภท
(classification) คือ class ที่ 4 (หรือ รถ
บรรทุก)

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Question

สมมติ จะแก้ปัญหา multi-class classification ที่มี 10 class ด้วย neural network มี 3 layers (ชั้น) และ hidden layer (layer 2) มี 5 units (หน่วย) 😊
(2) มีสมาชิกกี่ตัว

(i) 50

(ii) 55

(iii) 60

(iv) 66

$$s_{j+1} \times s_j$$

$$s_{j+1} \times (s_j + 1)$$

$$10 \times (5 + 1)$$

Question & Answer

สมมติ จะแก้ปัญหา multi-class classification ที่มี 10 class ด้วย neural network มี 3 layers (ชั้น) และ hidden layer (layer 2) มี 5 units (หน่วย) $\Theta^{(2)}$ มีสมาชิกกี่ตัว

(i) 50

(ii) 55

(iii) 60

(iv) 66

layer 2 มี 5 unit, layer 3 มี 10 unit

dimension ของ $\Theta^{(2)}$

$$= s_{j+1} \times (s_j + 1)$$

$$= s_3 \times (s_2 + 1)$$

$$= 10 \times (5 + 1)$$

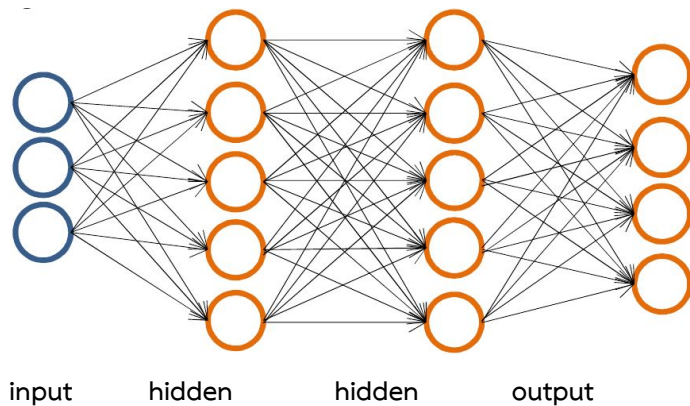
$$= 10 \times 6$$

ดังนั้น $\Theta^{(2)}$ มีสมาชิก 60 ตัว

สรุป: ความเข้าใจพื้นฐาน เกี่ยวกับ Neural network

ใน neural network เราอยากเรียนรู้ function $f: \mathbf{x} \mapsto \mathbf{y}$ เมื่อ

- สามารถมอง $f(\mathbf{x})$ เป็น neuron (เซลล์ประสาท) หรือ unit
- Neural network อาจเป็นแบบง่าย ก็คือ มีเพียง 1 neuron หรือ
- Neural network อาจเป็นแบบซับซ้อน โดย วาง หลากๆ units ซ้อนกัน เพื่อให้แต่ละ unit ส่ง output ไปให้อีก unit

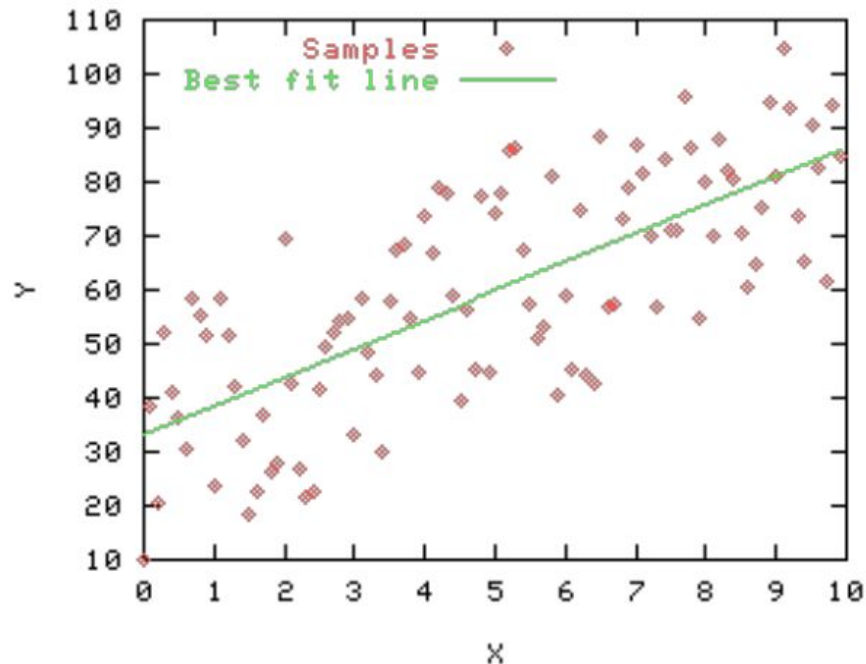


ความเข้าใจพื้นฐาน เกี่ยวกับ Neural network

- พยายามสร้าง neural network สำหรับ **univariate regression** (การถดถอยที่มี 1 ตัวแปร) setting
- ใน setting นี้ function $f: x \mapsto y$ ถูกแทนด้วย neuron 1 หน่วย ที่คำนวณ

$$f(x) = \max(ax + b, 0)$$

- เมื่อ a และ b เป็นสัมประสิทธิ์ที่มีค่าคงที่ (fixed coefficients) (ทำไมจึงสมเหตุสมผล ?)
- Unit แบบนี้เรียกว่า **ReLU (rectified linear unit)**

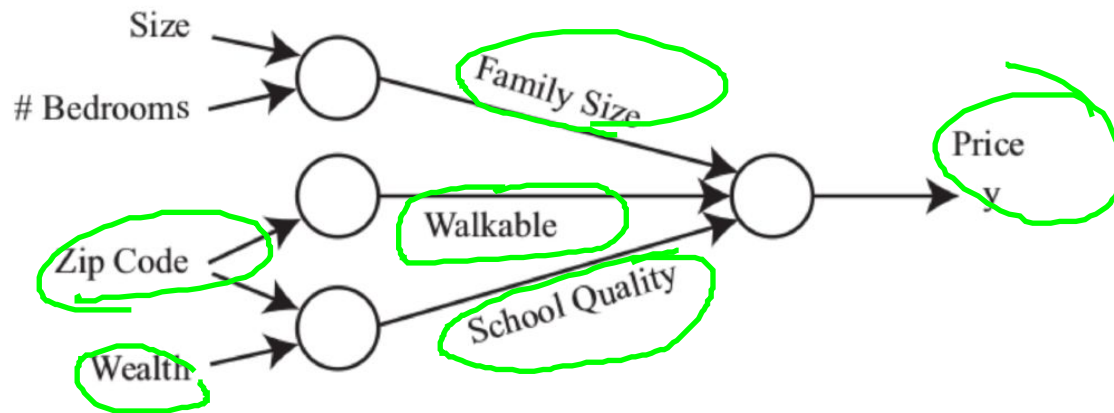


การเรียงซ้อน (stacking) unit เพื่อสร้าง function ที่ซับซ้อน

- จากตัวอย่างก่อนหน้านี้ จะเห็นว่า การเรียงซ้อน (stacking) หลายๆ unit สามารถสร้าง XNOR ได้
- พิจารณาตัวอย่างที่เป็นไปได้จริงมากขึ้น: เราอยากทำนายราคาบ้าน โดยใช้ข้อมูล: ขนาดพื้นที่บ้าน, จำนวนห้องนอน, รหัสไปรษณีย์ (postal code) และ ความร่ำรวยของบริเวณใกล้เคียงบ้าน
- เพื่อแก้ปัญหานี้ เราสามารถวางซ้อน (stack) หลายๆ unit ได้ดังนี้ :
 - คำนวณตัวแปร 'ขนาดครอบครัว' (family size) โดยอิงจากขนาดพื้นที่บ้าน และจำนวนห้องนอน
 - คำนวณว่าบริเวณใกล้เคียงบ้าน สามารถเดินได้ (walkable) มากเพียงไหน โดยอิงจากรหัสไปรษณีย์
 - คำนวณ 'คุณภาพโรงเรียน' (school quality) โดยอิงจากรหัสไปรษณีย์ และความร่ำรวยของบริเวณใกล้เคียงบ้าน

การเรียงซ้อน (stacking) unit เพื่อสร้าง function ที่ซับซ้อน

ท้ายสุด เราอาจตัดสินใจราคาบ้าน โดยขึ้นอยู่กับ features ที่หามา 3 ตัว ได้แก่ family size, walkable และ school quality



End-to-end Learning

เราเคยเห็นปัญหาที่ใช้ architecture นี้ที่อธิบาย?

เราไม่ต้องแก้ปัญหาลำนี้เอง เพราะการเรียนรู้ของ neural network เป็น **end-to-end learning**

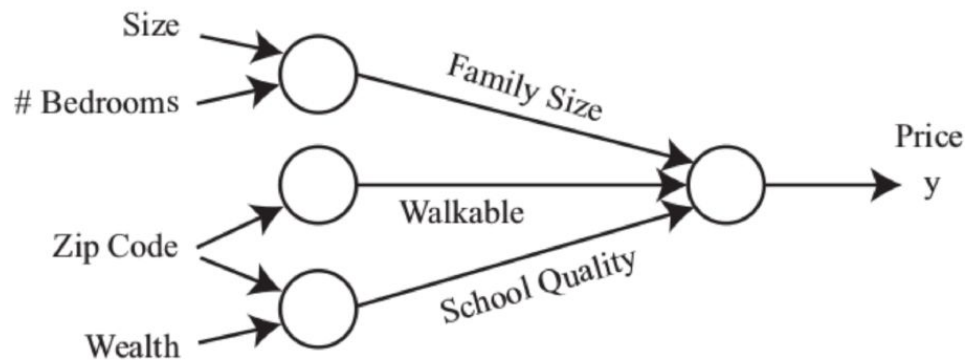
ซึ่งหมายความว่า **network** จะหาด้วยตัวเองว่า **intermediate features** (คุณลักษณะ) ใดที่ดีที่สุดสำหรับงานหรือปัญหาที่เราจะแก้

Hidden Units (Intermediate Units)

คือ neuron ที่อยู่ระหว่าง input และ output

สมมติ เรามี

- Input 4 ตัว: X_1, X_2, X_3, X_4
- Hidden units 3 หน่วย
- Output 1 ตัว: y



เป้าหมายของ network : หา intermediate features ที่จะทำนาย $y^{(j)}$ แต่ละตัว ได้ดีที่สุด โดยใช้ (จาก) $x^{(j)}$ ที่สอดคล้องกัน

เพราะความหมายของ intermediate features อาจเข้าใจได้ยาก neural network จึงถูกเรียกว่า **black box** (กล่องดำ หมายถึง สิ่งทำงานใต้งานหนึ่ง โดยผู้
ใช้ไม่จำเป็นต้องเข้าใจว่า สิ่งนั้นทำงานอย่างไร)

สรุป: Terminology (คำศัพท์เฉพาะทาง)

- **Neuron** หรือ **unit** ใช้ function บางอย่าง (เช่น ReLU, Sigmoid) กับ input เพื่อสร้าง output
- อาจเรียง unit ซ้อนกัน (stack) เพื่อสร้าง **neural networks**
- บางครั้งอาจอธิบาย/แทน input features ด้วย units เรียกว่า **input units** ซึ่งถูกรวบรวมเป็น **input layer**
- Output 1 หรือมากกว่า 1 ตัว รวมเป็น **output layer**
- Intermediate units เรียกว่า **hidden units** ซึ่งอาจถูกรวมเป็น **hidden layer ≥ 0 ชั้น**

References

1. Andrew Ng, Machine Learning, Coursera.
2. Teeradaj Racharak, AI Practical Development Bootcamp.