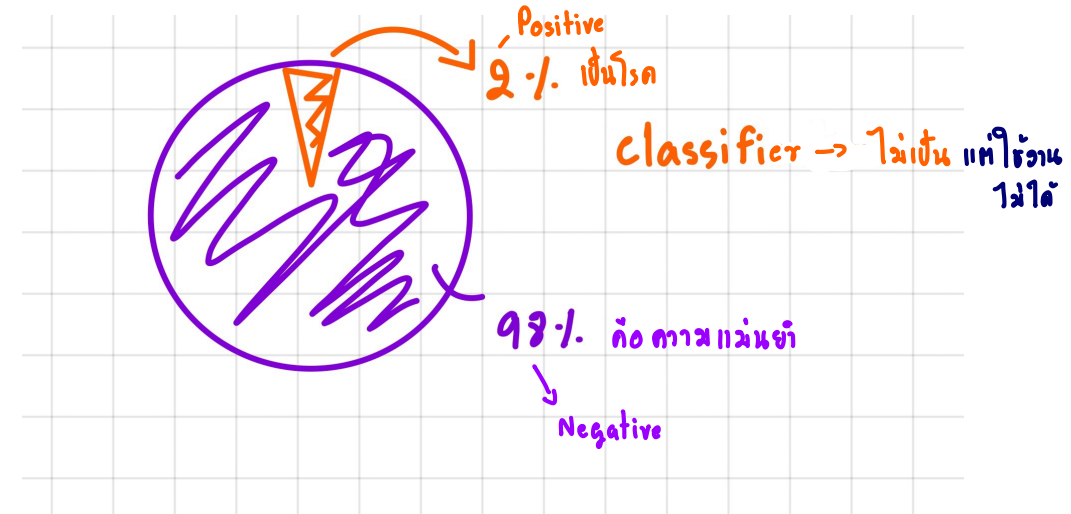


Model Evaluation and Selection

- Evaluation metrics
 - How can we measure accuracy?
 - Other metrics to consider?
- Use **validation test set** of class-labeled tuples instead of training set when assessing accuracy
- Methods for estimating a classifier's accuracy
 - Holdout method
 - Cross-validation
 - Bootstrap
- Comparing classifiers:
 - ROC Curves



Classifier Evaluation Metrics: Confusion Matrix

Confusion Matrix:

Actual class \ Predicted class	C_1	$\neg C_1$
C_1	True Positives (TP)	False Negatives (FN)
$\neg C_1$	False Positives (FP)	True Negatives (TN)

- In a confusion matrix w. m classes, $CM_{i,j}$ indicates # of tuples in class i that were labeled by the classifier as class j

- May have extra rows/columns to provide totals

Example of Confusion Matrix:

Actual class \ Predicted class	buy_computer = yes	buy_computer = no	Total
buy_computer = yes	6954	46	7000
buy_computer = no	412	2588	3000
Total	7366	2634	10000

Classifier Evaluation Metrics: Accuracy, Error Rate, Sensitivity and Specificity

A\P	C	¬C	
C	TP	FN	P
¬C	FP	TN	N
	P'	N'	All

❑ **Classifier accuracy**, or recognition rate

- ❑ Percentage of test set tuples that are correctly classified

ความแม่นยำ

$$\text{Accuracy} = (TP + TN) / \text{All}$$

❑ **Error rate**: $1 - \text{accuracy}$, or
 $\text{Error rate} = (FP + FN) / \text{All}$

❑ **Class imbalance problem**

- ❑ One class may be *rare*
 - ❑ E.g., fraud, or HIV-positive
- ❑ Significant *majority of the negative class* and minority of the positive class
- ❑ Measures handle the class imbalance problem
 - ❑ **Sensitivity** (recall): True positive recognition rate
 - ❑ $\text{Sensitivity} = TP / P$
 - ❑ **Specificity**: True negative recognition rate
 - ❑ $\text{Specificity} = TN / N$

Classifier Evaluation Metrics: Precision and Recall, and F-measures

- **Precision:** Exactness: what % of tuples that the classifier labeled as positive are actually positive?

$$P = \text{Precision} = \frac{TP}{TP + FP}$$

→ ค่าที่ model มองเป็น Positive ถูกต้องจริงแค่ไหน

- **Recall:** Completeness: what % of positive tuples did the classifier label as positive?

$$R = \text{Recall} = \frac{TP}{TP + FN}$$

→ model มองว่า
ค่าที่ model มองเป็น positive จริง
หรือไม่จริงแค่ไหน

- Range: [0, 1]
- The “inverse” relationship between precision & recall
- **F measure (or F-score):** harmonic mean of precision and recall
 - In general, it is the weighted measure of precision & recall

$$F_\beta = \frac{1}{\alpha \cdot \frac{1}{P} + (1 - \alpha) \cdot \frac{1}{R}} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

Assigning β times as much weight to recall as to precision)

- **F1-measure (balanced F-measure)**

- That is, when $\beta = 1$,

น้ำหนัก

$$F_1 = \frac{2PR}{P + R}$$

Preci → Recall

Chapter 9. Classification: Advanced Methods

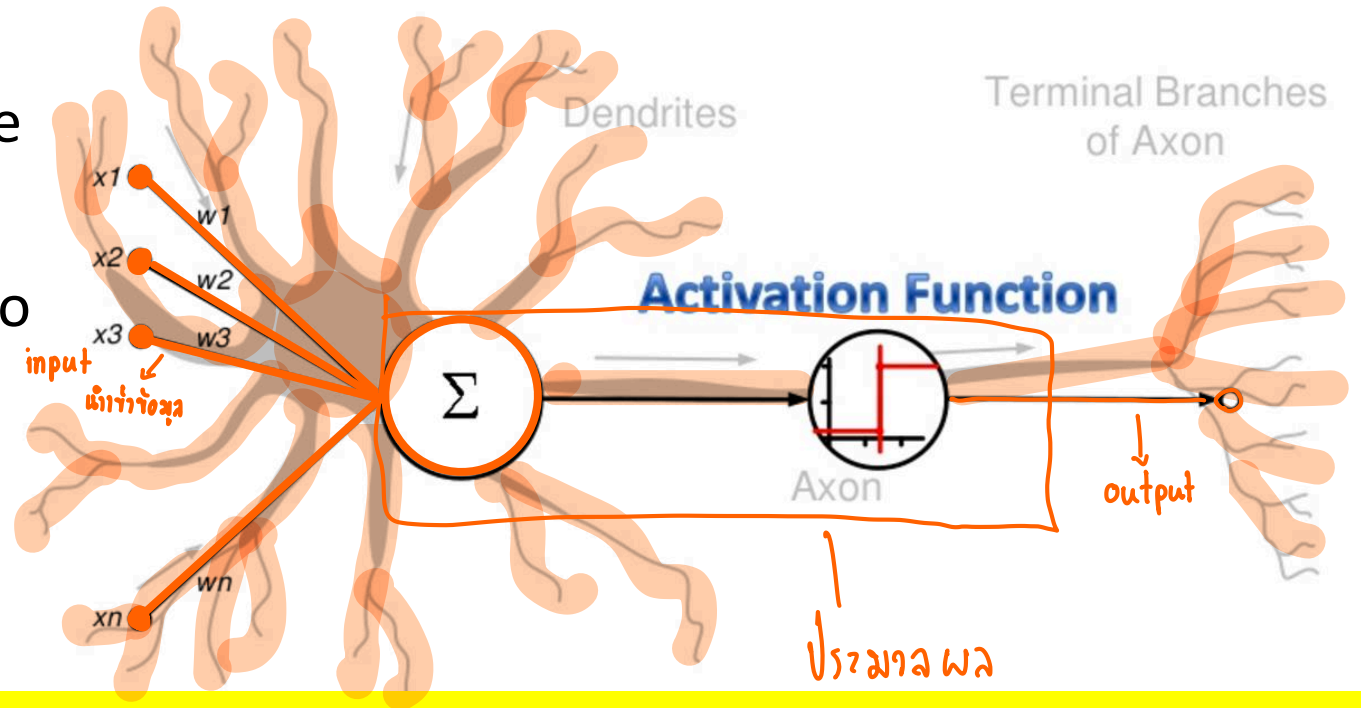
- ❑ Bayesian Belief Networks
- ❑ Support Vector Machines
- ❑ กลไกการตัดสินใจของ Neural Networks and Deep Learning
- ❑ Pattern-Based Classification
- ❑ Lazy Learners and K-Nearest Neighbors
- ❑ Other Classification Methods
- ❑ Summary



Artificial + Neural Network for Classification

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

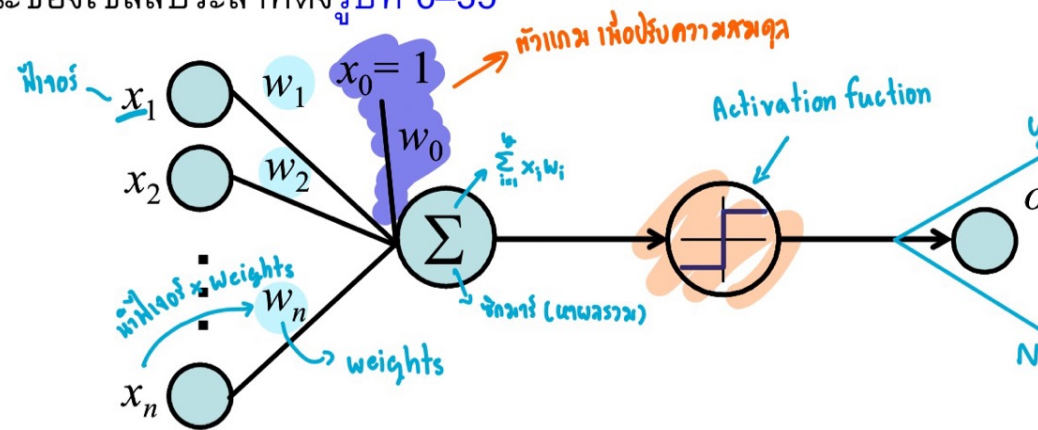
- ❑ Started by psychologists and neurobiologists to develop and test computational analogues of neurons
- ❑ A neural network: A set of connected input/output units where each connection has a **weight** associated with it
- ❑ During the learning phase, the **network learns by adjusting the weights** so as to be able to predict the correct class label of the input tuples



Artificial Neural Networks as an analogy of Biological Neural Networks

6.7.1 เพอร์เซปตรอน

เพอร์เซปตรอน (perceptron) เป็นข่ายงานประสาทเทียมแบบง่ายมีหน่วยเดียวที่จำลองลักษณะของเซลล์ประสาทดังรูปที่ 6-35



รูปที่ 6-35 เพอร์เซปตรอน

ฟังก์ชันกระตุ้น



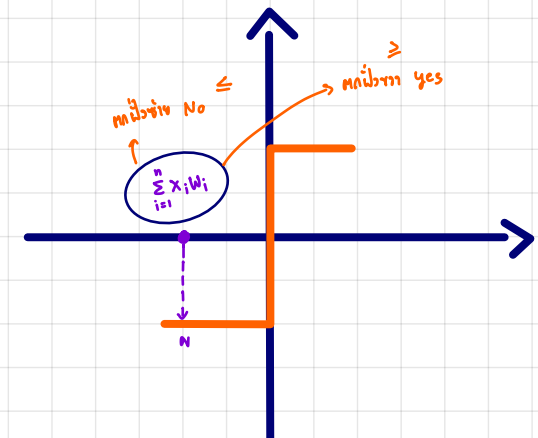
ในรูปแสดงฟังก์ชันกระตุ้น (activation function) ชนิดที่เรียกว่าฟังก์ชันสองขั้ว (bipolar function) ซึ่งแสดงผลของเอาต์พุตเป็น 1 กับ -1 ฟังก์ชันกระตุ้นอื่นๆ ที่นิยมใช้ก็อย่างเช่น ฟังก์ชันไบนารี (binary function) ซึ่งแสดงผลของเอาต์พุตเป็น 1 กับ 0 และเขียน



แทนด้วยรูป

เราสามารถแสดงเอาต์พุต (o) ในรูปของฟังก์ชันของอินพุต (x_1, x_2, \dots, x_n) ได้ดังนี้

$$o(x_1, x_2, \dots, x_n) = \begin{cases} 1 & \text{if } w_1x_1 + w_2x_2 + \dots + w_nx_n > \theta \\ -1 & \text{if } w_1x_1 + w_2x_2 + \dots + w_nx_n < \theta \end{cases} \quad (6.7)$$



ตารางที่ 6-17 อัลกอริทึมการเรียนรู้เพอร์เซปตรอน

Algorithm: Perceptron-Learning-Rule

ตอบถูกหมด (นุช)

ตรวจสอบ
นุช

1. Initialize weights w_i of the perceptron. *← กำหนด w_i มาให้*
2. UNTIL the termination condition is met DO *← รับ Data มาให้*
 - 2.1 FOR EACH training example DO *← $f(\sum w_i x_i)$*
 - Input the example and compute the output.
 - Change the weights if the output from the perceptron is not equal to the target output using the following rule.

$$w_i \leftarrow w_i + \Delta w_i$$

weight ใหม่ *weight เดิม* *penalty*

$$\Delta w_i \leftarrow \alpha (t - o) x_i$$

Learning rate *output ของ model ที่ออกมา* *input x_0, x_1, x_2* *0.5 target-label ที่ต้อง*

where t , o and α are the target output, the output from the perceptron and the learning rate, respectively.

6.7.2 ตัวอย่างการเรียนรู้ฟังก์ชัน AND และ XOR ด้วยกฎการเรียนรู้เพอร์เซปตรอน

พิจารณาตัวอย่างการเรียนรู้ของเพอร์เซปตรอนโดยจะให้เรียนรู้ฟังก์ชัน 2 ฟังก์ชัน ฟังก์ชันแรกคือฟังก์ชัน AND แสดงในตารางที่ 6-18 ในกรณีนี้เราใช้ฟังก์ชันไบนารีเป็นฟังก์ชันกระตุ้น

X **Y**

ตารางที่ 6-18 ฟังก์ชัน AND(x_1, x_2)

	x_1	x_2	เอาต์พุตเป้าหมาย
1	0	0	0
2	0	1	0
3	1	0	0
4	1	1	1

$T \wedge T \equiv T$
 $T \wedge F \equiv F$
 $F \wedge T \equiv F$
 $F \wedge F \equiv F$

ฟังก์ชัน AND ตามตารางด้านบนนี้จะให้ค่าที่เป็นจริงก็ต่อเมื่อ x_1 และ x_2 เป็นจริงทั้งคู่ (ดูที่สแตมภ์เอาต์พุตเป้าหมาย) ผลการใช้กฎการเรียนรู้เพอร์เซปตรอนกับฟังก์ชัน AND แสดงในตารางที่ 6-19

ตารางที่ 6-19 ผลการเรียนรู้ฟังก์ชัน AND โดยกฎการเรียนรู้เพอร์เซปตรอน

✂️ ข้อสอบออกเกินตาราง

1 epoch
1 รอบ

Perceptron Learning Example - Function AND												
		Bias Input $x_0=+1$				$\sum_{i=0}^n (x_i w_i)$	$f(\sum_{i=0}^n (x_i w_i))$	Alpha = 0.5	Learning rate	Weight Values		
Input	Input				Net Sum	Target	Actual	Alpha*		w0	w1	w2
x1	x2	$1.0*w_0$	$x1*w_1$	$x2*w_2$	Input	Output	Output	Error				
										0.1	0.1	0.1
0	0	0.10	0.00	0.00	0.10	0	1	-0.50		-0.40	0.10	0.10
0	1	-0.40	0.00	0.10	-0.30	0	0	0.00		-0.40	0.10	0.10
1	0	-0.40	0.10	0.00	-0.30	0	0	0.00		-0.40	0.10	0.10
1	1	-0.40	0.10	0.10	-0.20	1	0	0.50		0.10	0.60	0.60
0	0	0.10	0.00	0.00	0.10	0	1	-0.50		-0.40	0.60	0.60
0	1	-0.40	0.00	0.60	0.20	0	1	-0.50		-0.90	0.60	0.10
1	0	-0.90	0.60	0.00	-0.30	0	0	0.00		-0.90	0.60	0.10
1	1	-0.90	0.60	0.10	-0.20	1	0	0.50		-0.40	1.10	0.60
0	0	-0.40	0.00	0.00	-0.40	0	0	0.00		-0.40	1.10	0.60
0	1	-0.40	0.00	0.60	0.20	0	1	-0.50		-0.90	1.10	0.10
1	0	-0.90	1.10	0.00	0.20	0	1	-0.50		-1.40	0.60	0.10
1	1	-1.40	0.60	0.10	-0.70	1	0	0.50		-0.90	1.10	0.60
0	0	-0.90	0.00	0.00	-0.90	0	0	0.00		-0.90	1.10	0.60
0	1	-0.90	0.00	0.60	-0.30	0	0	0.00		-0.90	1.10	0.60
1	0	-0.90	1.10	0.00	0.20	0	1	-0.50		-1.40	0.60	0.60
1	1	-1.40	0.60	0.60	-0.20	1	0	0.50		-0.90	1.10	1.10
0	0	-0.90	0.00	0.00	-0.90	0	0	0.00		-0.90	1.10	1.10
0	1	-0.90	0.00	1.10	0.20	0	1	-0.50		-1.40	1.10	0.60
1	0	-1.40	1.10	0.00	-0.30	0	0	0.00		-1.40	1.10	0.60
1	1	-1.40	1.10	0.60	0.30	1	1	0.00		-1.40	1.10	0.60
0	0	-1.40	0.00	0.00	-1.40	0	0	0.00		-1.40	1.10	0.60
0	1	-1.40	0.00	0.60	-0.80	0	0	0.00		-1.40	1.10	0.60
1	0	-1.40	1.10	0.00	-0.30	0	0	0.00		-1.40	1.10	0.60
1	1	-1.40	1.10	0.60	0.30	1	1	0.00		-1.40	1.10	0.60

$$0.1 + 0.5(0-1)0 = 0.1$$

$$0.1 + 0.5(0-1)1 = 0.1 + (-0.5) = -0.4$$

$$0.1 + 0.5(1-0)1 = 0.6$$

$$0.1 + 0.5(1-0)1 = 0.6$$

$$-0.4 + 0.5(1-0)1 = 0.1$$