# Chapter 9. Classification: Advanced Methods

- Bayesian Belief Networks
- Support Vector Machines
- □ Neural Networks and Deep Learning



- Pattern-Based Classification
- ☐ Lazy Learners and K-Nearest Neighbors
- Other Classification Methods
- Summary



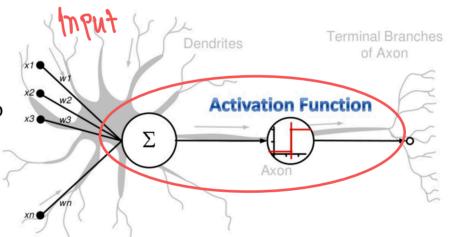
# Antificial Neural Network for Classification

□ 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

## Discussion on the **k**-NN Algorithm

- □ k-NN for <u>real-valued prediction</u> for a given unknown tuple ! คำหางว่านานกุรัณรักน

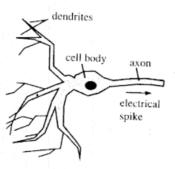
  - Returns the mean values of the k nearest neighbors
- Distance-weighted nearest neighbor algorithm
  - Weight the contribution of each of the k neighbors according to their distance to the query  $x_q$
  - ☐ Give greater weight to closer neighbors

$$w \equiv \frac{1}{d(x_q, x_i)^2}$$

- □ Robust to noisy data by averaging *k*-nearest neighbors
- Curse of dimensionality: distance between neighbors could be dominated by irrelevant attributes
  - To overcome it, axes stretch or elimination of the least relevant attributes

## 6.7 ข่ายงานประสาทเทียม

ข่ายงานประสาทเทียม (Artificial Neural Network) เป็นการจำลองการทำงานบางส่วนของ สมองมนุษย์ เซลล์ประสาท (neuron) ในสมองของคนเราประกอบด้วยนิวเคลียส (nucleus) ตัวเซลล์ (cell body) ใยประสาทนำเข้า (dendrite) แกนประสาทนำออก (axon) แสดงใน รูปที่ 6–34

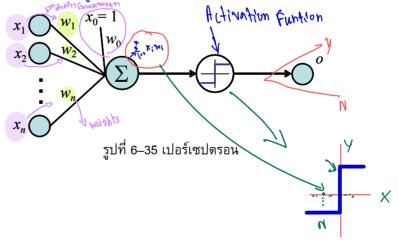


รูปที่ 6–34 เซลล์ประสาท

เดนใดรท์ทำหน้าที่รับสัญญาณไฟฟ้าเคมีซึ่งส่งมาจากเซลล์ประสาทใกล้เคียง เซลล์ ประสาทตัวหนึ่งๆ จะเชื่อมต่อกับเซลล์ตัวอื่นๆ ประมาณ 10,000 ตัว เมื่อสัญญาณไฟฟ้าเคมี ที่รับเข้ามาเกินค่าค่าหนึ่ง เซลล์จะถูกกระตุ้นและส่งสัญญาณไปทางแกนประสาทนำออกไป ยังเซลล์อื่นๆ ต่อไป ประมาณกันว่าสมองของคนเรามีเซลล์ประสาทอยู่ทั้งสิ้นประมาณ 10<sup>11</sup> ตัว

#### 6.7.1 เพอร์เซปตรอน

เพอร์เซปตรอน (perceptron) เป็นข่ายงานประสาทเทียมแบบง่ายมีหน่วยเดียวที่จำลอง ลักษณะของเซลล์ประสาทดังรูปที่ 6–35



0 0

ตารางที่ 6–17 อัลกอริทึมกฏการเรียนรู้เพอร์เซปตรอน

Algorithm: Perceptron-Learning-Rule

1. Initialize weights woof the perceptron. ผู้การบาง อิจโล ได้ตัว \* เทโป ตำและ (รนค)
2. UNTIL the termination condition is met DO

2.1 FOR EACH training example DO

- Input the example and compute the output. f(4wx)

- Change the weights if the output from the perceptron is not equal to the target output using the following rule.

Xo Xy X

where t, o and  $\alpha$  are the target output, the output from the perceptron and the learning rate, respectively.

การปรับน้ำหนักตามกฎการเรียนรู้เพอร์เซปตรอนโดยใช้อัตราการเรียนรู้ที่มีค่าน้อย เพียงพอ จะได้ระนาบหลายมิติที่จะลู่เข้าสู่ระนาบหนึ่งที่สามารถแบ่งข้อมูลออกเป็นสองส่วน (ในกรณีที่ข้อมูลสามารถแบ่งได้) เพื่ออธิบายผลที่เกิดจากการปรับค่าน้ำหนัก เราจะลอง พิจารณาพฤติกรรมของกฏการเรียนรู้นี้ดูว่าทำไมการปรับน้ำหนักเช่นนี้จึงลู่เข้าสู่ระนาบที่ แบ่งข้อมูลได้อย่างถูกต้อง

- พิจารณากรณีแรกที่เพอร์เซปตรอนแยกตัวอย่างสอนตัวหนึ่งที่รับเข้ามาได้ถูกต้อง กรณีนี้จะพบว่า (t-o) ดังนั้น  $\Delta w_i$  ไม่เปลี่ยนแปลงเพราะ จะมีค่าเป็น 0  $\Delta w_i = \alpha(t-o)x_i$
- พิจารณาในกรณีที่เพอร์เซปตรอนให้เอาต์พุตเป็น –1 แต่เอาต์พุตเป้าหมายหรือ ค่าที่แท้จริงเท่ากับ 1 ในกรณีนี้หมายความว่าค่าที่เราต้องการคือ 1 แต่ค่าน้ำหนัก ไม่เหมาะสม ดังนั้นเพื่อที่จะทำให้เพอร์เซปตรอนให้เอาต์พุตเป็น 1 น้ำหนักต้องถูก ปรับให้สามารถเพิ่มค่าของ  $ec{w}\cdotec{x}$  ในกรณีนี้หมายความว่าผลรวมเชิงเส้นน้อย เกินไปและน้อยกว่า 0 จึงได้เอาต์พุตเป็น -1 ดังนั้นสิ่งที่เราต้องการคือการเพิ่มค่า ผลรวมเชิงเส้นเพราะถ้าเราเพิ่มค่าได้เรื่อยๆ จนมากกว่า 0 เพอร์เซปตรอนจะให้ เอาต์พุตเป็น 1 ซึ่งตรงกับที่เราต้องการ พิจารณาดูดังต่อไปนี้ว่าการปรับค่าโดยกฎ เรียนรู้ทำให้ผลรวมเชิงเส้นเพิ่มขึ้นได้อย่างไร กรณีนี้เราจะได้ว่า (t-o) เท่ากับ (1-(-1)) มีค่าเป็น 2 และลองพิจารณาค่าของอินพุต  $x_i$  แยกกรณีดังนี้

- $\circ$  ถ้า  $x_i > 0$  จะได้ว่า  $\Delta w_i$  มากกว่า 0 เพราะว่า  $\Delta w_i \leftarrow \alpha(t-o)x_i$  และ  $\alpha$  มากกว่า 0, (t-o) = 2 และ  $x_i > 0$  จากสมการการปรับน้ำหนัก  $w_i \leftarrow w_i + \Delta w_i$  เมื่อ  $\Delta w_i$  มากกว่า 0 จะทำให้  $w_i$  มีค่าเพิ่มขึ้นและ  $\sum w_i x_i$  ก็จะมีค่าเพิ่มขึ้น เมื่อผลรวมมีค่ามากขึ้นแสดงว่าการปรับไปในทิศทางที่ ถูกต้องคือเมื่อปรับไปจนกระทั่งได้ผลรวมมากกว่า 0 จะทำให้ เพอร์เซปตรอนเอาต์พุตได้ถูกต้องยิ่งขึ้น
- $\circ$  ถ้า  $x_i < 0$  เราจะได้ว่า  $\alpha(t-o)x_i$  จะมีค่าน้อยกว่า 0 แสดงว่า  $w_i$  ตัวที่คูณ กับ  $x_i$  ที่น้อยกว่า 0 จะลดลงทำให้  $\sum w_i x_i$  เพิ่มขึ้นเหมือนเดิม เพราะ  $x_i$  เป็นค่าลบและ  $w_i$  มีค่าลดลง ในที่สุดก็จะทำให้เพอร์เซปตรอนให้ เอาต์พุตได้ถูกต้องยิ่งขึ้น
- ในกรณีที่เพอร์เซปตรอนให้เอาต์พุตเป็น 1 แต่เอาต์พุตเป้าหมายหรือค่าที่แท้จริง เท่ากับ -1 จะได้ว่า  $w_i$  ของ  $x_i$  ที่เป็นค่าบวกจะลดลง ส่วน  $w_i$  ของ  $x_i$  ที่เป็นค่าลบ จะเพิ่มขึ้นและทำให้การปรับเป็นไปในทิศทางที่ถูกต้องเช่นเดียวกับในกรณีแรก

### 6.7.2 ตัวอย่างการเรียนฟังก์ชัน AND และ XOR ด้วยกฎเรียนรู้เพอร์เซปตรอน

พิจารณาตัวอย่างการเรียนรู้ของเพอร์เซปตรอนโดยจะให้เรียนรู้ฟังก์ชัน 2 ฟังก์ชัน ฟังก์ชัน แรกคือฟังก์ชัน AND แสดงในตารางที่ 6–18 ในกรณีนี้เราใช้ฟังก์ชันไบนารีเป็นฟังก์ชัน กระตุ้น

•		
ุตารางที่ 6–	AND(x1,x2)	
$x_1$	$x_2$	เอาต์พุต
		เป้าหมาย
0	0	0
0	1	0
1	0	0
1	1	1
	0	0 0

ฟังก์ชัน AND ตามตารางด้านบนนี้จะให้ค่าที่เป็นจริงก็ต่อเมื่อ x1 และ x2 เป็นจริงทั้งคู่ (ดูที่ สดมภ์เอาต์พุดเป้าหมาย) ผลการใช้กฎการเรียนรู้เพอร์เซปตรอนกับฟังก์ชัน AND แสดงใน ตารางที่ 6–19

שמשים שומחון אנחשים שמשים

ตารางที่ 6–19 ผลการเรียนรู้ฟังก์ชัน AND โดยกฎการเรียนรู้เพอร์เซปตรอน													
	Perceptron Learning Example - Function AND												
						Cix	4) _	_, fl	E LKI	41	7/00	- 0	^
<b>10</b>			Bias Inpu	ıt x0=+1		3,0	Y	Alpha =	0.5	1	1	1	anto scom
	Input	Input	V V			Net Sum		Actual	Alpha*	W	eight Valu	ies	0.1+0 9
1 6 ×	xl	x2	1.0*w0	x1*w1	x2*w2	= Input	Output	Qutput	Error	wo	w	w2	,
'\										0.1	0.1	0.1	<b>←</b> +076 <sup>8</sup>
100 S	0	0	0.10	<b>4</b> 0.00	<b>%</b> 0.00	0.10	0		-0.50	-0.40	0.10	0.10	W. 101+ 1-00-01
1100 5	0	1	-0.40	0.00	<u>+</u> 0.10	-0.30	0	0	6.00	-0.40	0.10	0.10	01 +C-05
100) {	1	0	-0.40	<b>→</b> 0.10	<b>†</b> 0.00	-0.30	0	0	<b>→</b> 0.00	-0.40	0.10	0.10	Turet - Actual
C	1	1	- <u>0.40</u>	<u>+</u> 0.10		-0.20	1	0	0.50	0.10	79.60	0.60	(00,01
- (	0	0		0.00	0.00	0.10	O	1	-0.50	-0.40	0.60	0.60	0.1 +05(1-0)1
( CO)	0	1	-0.40	0.00	0.60	0.20	0	1	-0.50	-0.90	0.60	0.10	0.4 * 0.5 (4-0)4
7	1	0	-0.90	0.60	0.00	-0.30	0	0		-0.90	0.60	0.10	0.1
	1	1	-0.90	0.60	0.10	-0.20	1	0		-0.40	1.10	0.60	
	0	0		0.00	0.00	-0.40	0	0	/ 0.0	-0.40	1.10	0.60	-0.4 0.5(4-0)1
	0	1	-0.40	0.00	0.60	0.20	0	1	-0.50	-0.90	1.10	0.10	_ 0.a
	1	0	-0.90 -1.40	1.10 0.60	0.00	-0.70	0	1	-0.50 0.50	-1.40 -0.90	0.60 1.10	0.10 0.60	
	1	0	-0.90	0.00	0.10	-0.70	0	0		-0.90	1.10	0.60	
	0	1	-0.90	0.00	0.60	-0.30	0	0		-0.90	1.10	0.60	
	1	0		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		-0.90	1.10	1.10	
	0	0		0.00	0.00	-0.90	0	0	$\sim$	-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	9.00	-1.40	1.10	0.60	
1.6	0	0	-1.40	0.00	0.00	-1.40	0	0	0.00	-1.40	1.10	0.60	9 0 = 1
<b>⊝=</b> 0	0	1	-1.40	0.00	0.60	-0.80	0	0	0.00	-1.40	1.10	0.60	4871
{2} — (I)→ output	1	0	-1.40	1.10	0.00	-0.30	0	0	0.00	-1.40	1.10	0.60	
$\gamma$	1	1	-1.40	1.10	0.60	0.30	1	1	0.00	-1.40	1.10	0.60	J

ขั้นตอนแรกเริ่มจากการสุ่มค่า  $w_0$  จนถึง  $w_2$  ในที่นี้กำหนดให้เป็น  $0.1\,$  ทั้งสามตัว จากนั้น ก็เริ่มป้อนตัวอย่างเข้าไป (ทีละแถว) ตัวอย่างแรกได้ผลรวมเชิงเส้น (Net Sum) เป็น 0.10 ซึ่งมากกว่า 0 ดังนั้นเปอร์เซปตรอนจะให้เอาต์พุตจริง (Actual Output) ออกมาเป็น 1 ซึ่งผิด เพราะเอาต์พุตเป้าหมาย (Target Output) จะต้องได้เป็น 0 ทำให้อัตราการเรียนรู้คูณค่า ผิดพลาด (Alpha x Error) ได้ –0.50 หลังจากนี้ก็นำไปปรับน้ำหนักตาม  $w_i \leftarrow w_i + \Delta w_i$  และ  $\Delta w_i \leftarrow \alpha(t-o)x_i$  ดังนั้นจะได้เป็น  $w_0 \leftarrow w_0 + \alpha(t-o)x_0 = w_0 + 0.50(-1) \times 1 = 0.10 + (-0.5)$ = -0.4 ต่อไปก็ปรับค่า  $w_1$  ในทำนองเดียวกัน  $w_1 \leftarrow w_1 + \alpha(t-o)x_1$  =  $w_1$  + 0.50(-1) x 0 ดังนั้น  $w_1$  จะเท่ากับ 0.10 คือไม่เปลี่ยนแปลง เช่นเดียวกับ  $w_2$  ที่ไม่เปลี่ยนแปลง จะเห็นได้ ว่าแม้มีค่าผิดพลาดแต่ไม่มีการปรับค่า  $w_1$  และ  $w_2$  เนื่องจากอินพุตที่ใส่เข้าไปเป็น 0 ทำ