
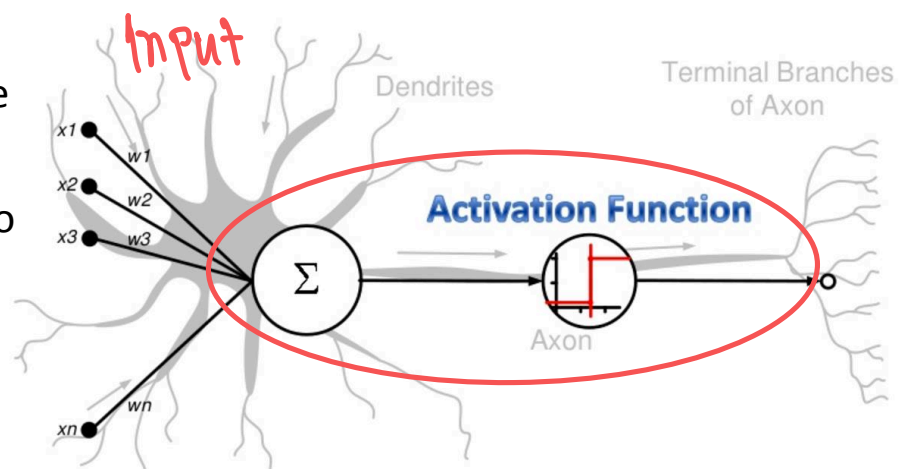


Chapter 9. Classification: Advanced Methods

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

Artificial + **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

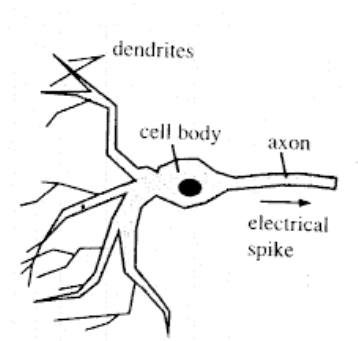
- k -NN for real-valued prediction for a given unknown tuple
 - Returns the mean values of the k nearest neighbors
- Distance-weighted nearest neighbor algorithm
 - Weight the contribution of each of the k neighbors according to their distance to the query x_q
 - Give greater weight to closer neighbors
- 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

1. หาจำนวนเพื่อนบ้าน
2. คำนวณค่าเฉลี่ยเพื่อนบ้าน
3. คำนวณค่าเฉลี่ยเพื่อนบ้าน

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

6.7 ข่ายงานประสาทเทียม

ข่ายงานประสาทเทียม (Artificial Neural Network) เป็นการจำลองการทำงานบางส่วนของสมองมนุษย์ เซลล์ประสาท (neuron) ในสมองของคนเราประกอบด้วยนิวเคลียส (nucleus) ตัวเซลล์ (cell body) โยประสาทนำเข้า (dendrite) แกนประสาทนำออก (axon) แสดงในรูปที่ 6-34

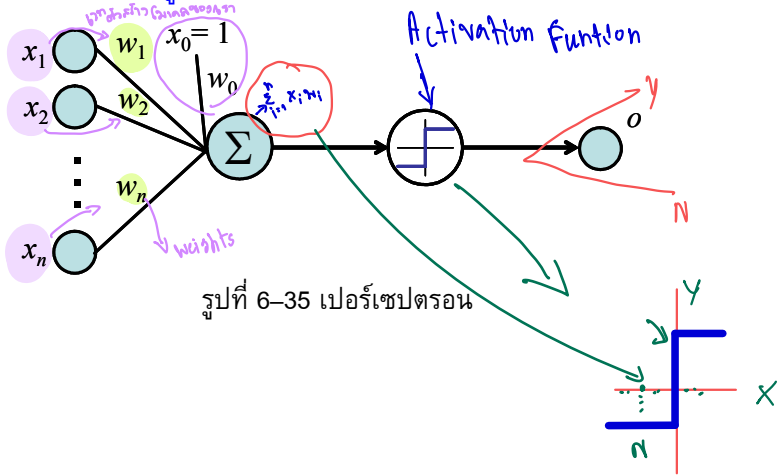


รูปที่ 6-34 เซลล์ประสาท

เดนไดรต์ทำหน้าที่รับสัญญาณไฟฟ้าเคมีซึ่งส่งมาจากเซลล์ประสาทใกล้เคียง เซลล์ประสาทตัวหนึ่งๆ จะเชื่อมต่อกับเซลล์ตัวอื่นๆ ประมาณ 10,000 ตัว เมื่อสัญญาณไฟฟ้าเคมีที่รับเข้ามาเกินค่าหนึ่ง เซลล์จะถูกกระตุ้นและส่งสัญญาณไปทางแกนประสาทนำออกไปยังเซลล์อื่นๆ ต่อไป ประมาณกันว่าสมองของคนเรามีเซลล์ประสาทอยู่ทั้งสิ้นประมาณ 10^{11} ตัว

6.7.1 เพอร์เซปตรอน

เพอร์เซปตรอน (perceptron) เป็นข่ายงานประสาทเทียมแบบง่ายมีหน่วยเดียวที่จำลองลักษณะของเซลล์ประสาทดังรูปที่ 6-35



รูปที่ 6-35 เพอร์เซปตรอน

- ถ้า $x_i > 0$ จะได้ว่า Δw_i มากกว่า 0 เพราะว่า $\Delta w_i \leftarrow \alpha(t-o)x_i$ และ α มากกว่า 0, $(t-o) = 2$ และ $x_i > 0$ จากสมการการปรับน้ำหนัก $w_i \leftarrow w_i + \Delta w_i$ เมื่อ Δw_i มากกว่า 0 จะทำให้ w_i มีค่าเพิ่มขึ้นและ $\sum w_i x_i$ ก็จะมีค่าเพิ่มขึ้น เมื่อผลรวมมีค่ามากขึ้นแสดงว่าการปรับไปในทิศทางที่ถูกต้องคือเมื่อปรับไปจนกระทั่งได้ผลรวมมากกว่า 0 จะทำให้เพอร์เซปตรอนเอาต์พุตได้ถูกต้องยิ่งขึ้น
- ถ้า $x_i < 0$ เราจะได้ว่า $\alpha(t-o)x_i$ จะมีค่าน้อยกว่า 0 แสดงว่า w_i ตัวที่คูณกับ x_i ที่น้อยกว่า 0 จะลดลงทำให้ $\sum w_i x_i$ เพิ่มขึ้นเหมือนเดิม เพราะ x_i เป็นค่าลบและ w_i มีค่าลดลง ในที่สุดก็จะทำให้เพอร์เซปตรอนให้เอาต์พุตได้ถูกต้องยิ่งขึ้น
- ในกรณีที่เพอร์เซปตรอนให้เอาต์พุตเป็น 1 แต่เอาต์พุตเป้าหมายหรือค่าที่แท้จริงเท่ากับ -1 จะได้ว่า w_i ของ x_i ที่เป็นค่าบวกจะลดลง ส่วน w_i ของ x_i ที่เป็นค่าลบจะเพิ่มขึ้นและทำให้การปรับเป็นไปในทิศทางที่ถูกต้องเช่นเดียวกับในกรณีแรก

6.7.2 ตัวอย่างการเรียนรู้ฟังก์ชัน AND และ XOR ด้วยกฎเรียนรู้เพอร์เซปตรอน

พิจารณาตัวอย่างการเรียนรู้ของเพอร์เซปตรอนโดยจะให้เรียนรู้ฟังก์ชัน 2 ฟังก์ชัน ฟังก์ชันแรกคือฟังก์ชัน AND แสดงในตารางที่ 6-18 ในกรณีนี้เราใช้ฟังก์ชันไบนารีเป็นฟังก์ชันกระตุ้น

ตารางที่ 6-18 ฟังก์ชัน AND(x1,x2)

x1	x2	เอาต์พุตเป้าหมาย
0	0	0
0	1	0
1	0	0
1	1	1

X

Y

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

ฟังก์ชัน AND ตามตารางด้านบนนี้จะให้ค่าที่เป็นจริงก็ต่อเมื่อ x1 และ x2 เป็นจริงทั้งคู่ (ดูที่สตรัมภ์เอาต์พุตเป้าหมาย) ผลการใช้กฎการเรียนรู้เพอร์เซปตรอนกับฟังก์ชัน AND แสดงในตารางที่ 6-19

ตารางที่ 6-19 ผลการเรียนรู้ฟังก์ชัน AND โดยกฎการเรียนรู้เพอร์เซปตรอน

Perceptron Learning Example - Function AND												
		Bias Input x0=+1					Alpha = 0.5		Weight Values			
Input	Input	1.0*w0	x1*w1	x2*w2	Net Sum	Target	Actual	Alpha*	Error	w0	w1	w2
x1	x2				Input	Output	Output	Error				
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	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	

ขั้นตอนแรกเริ่มจากการสุ่มค่า w_0 จนถึง w_2 ในที่นี้กำหนดให้เป็น 0.1 ทั้งสามตัว จากนั้นก็เริ่มป้อนตัวอย่างเข้าไป (ทีละแถว) ตัวอย่างแรกได้ผลรวมเชิงเส้น (Net Sum) เป็น 0.10 ซึ่งมากกว่า 0 ดังนั้นเปอร์เซปตรอนจะให้เอาต์พุตจริง (Actual Output) ออกมาเป็น 1 ซึ่งผิดเพราะเอาต์พุตเป้าหมาย (Target Output) จะต้องได้เป็น 0 ทำให้อัตราการเรียนรู้คูณค่าผิดพลาด (Alpha x Error) ได้ -0.50 หลังจากนั้นก็นำไปปรับน้ำหนักตาม $w_i \leftarrow w_i + \Delta w_i$ 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 $w_1 \leftarrow w_1 + \alpha(t-o)x_1 = w_1 + 0.50(-1) \times 0$ ดังนั้น w_1 จะเท่ากับ 0.10 คือไม่เปลี่ยนแปลง เช่นเดียวกับ w_2 ที่ไม่เปลี่ยนแปลง จะเห็นได้ว่าแม้มีค่าผิดพลาดแต่ไม่มีการปรับค่า w_1 และ w_2 เนื่องจากอินพุตที่ใส่เข้าไปเป็น 0 ทำ