

523414:

Artificial Neural Networks (ANN)

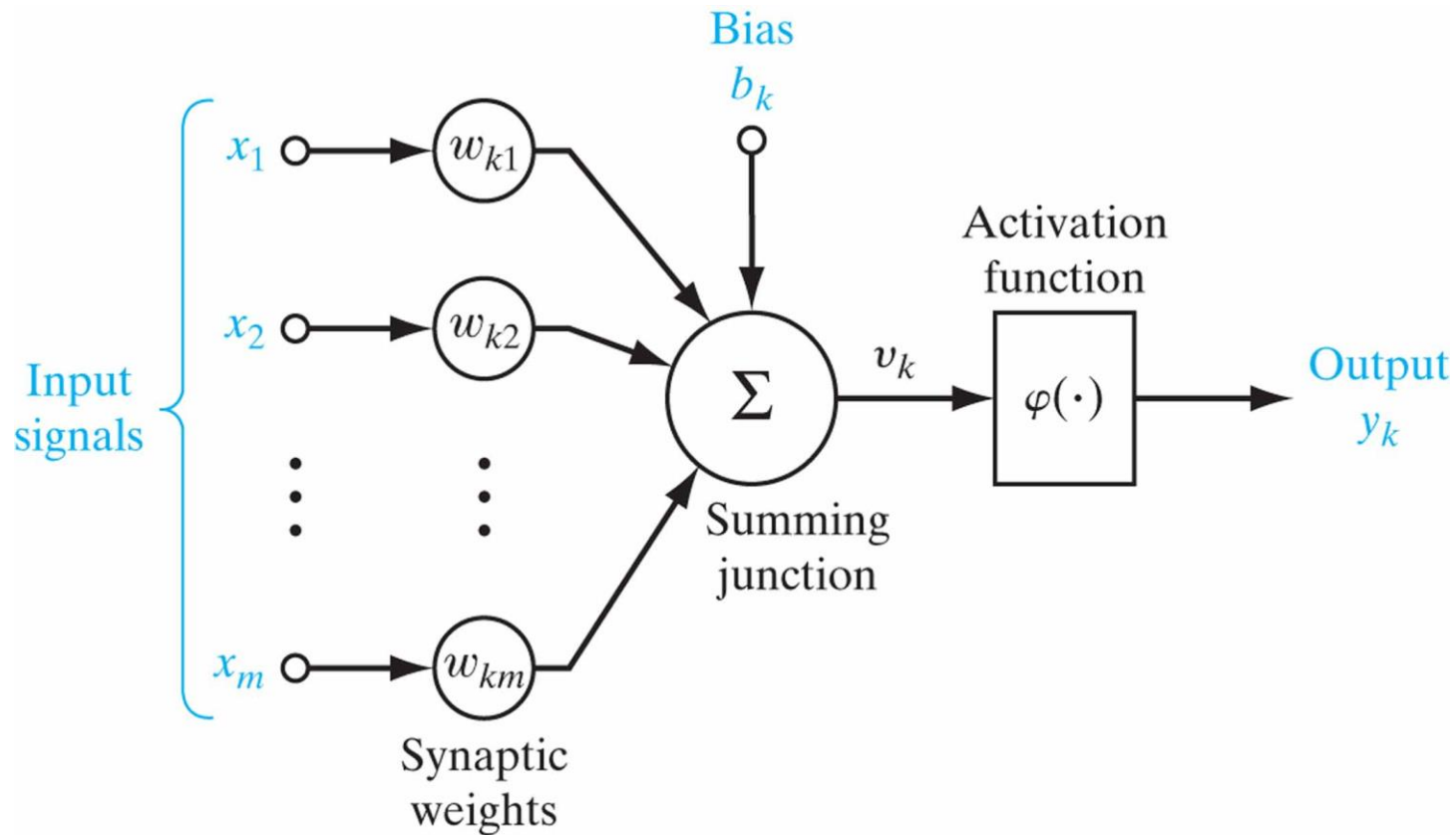
Lecture 01:

Artificial Neural Networks Overview

**Institute of Engineering
School of Computer Engineering
Suranaree University of Technology**

Artificial Neural Networks (ANN)?

Model of a Neuron



Overview of Artificial Neural Networks

- เครือข่ายประสาทเทียม (Artificial Neural Network หรือ ANN) เป็นเครือข่ายที่มีรูปแบบโครงสร้างและการทำงานของการทำงานประมาณผลเหมือนกับสมองในสิ่งมีชีวิตที่ซึ่งมีปรับเปลี่ยนตัวเองต่อการตอบสนองของอินพุตตามกฎของการเรียนรู้ (learning rule) หลังจากที่เครือข่ายได้เรียนรู้สิ่งที่ต้องการแล้ว เครือข่ายนั้นจะสามารถทำงานที่กำหนดไว้ได้
- ตัวเครือข่ายทำการเก็บข้อมูลความรู้ (knowledge) ในระหว่างขั้นตอนของการเรียนรู้ โดยทำการเก็บค่าไว้ที่น้ำหนักประสาท (synaptic weights) ซึ่งโครงสร้างของตัวนิวรอนภายในเครือข่ายนั้นมีอยู่มากมายหลายชนิด
- โครงสร้างดังกล่าวเป็นองค์ประกอบสำคัญที่ทำให้คุณลักษณะต่างๆ ของเครือข่ายแตกต่างกันออกไป ไม่ว่าจะเป็นการจัดวางเรียงตัวของนิวรอน กฎการเรียนรู้ที่ทำให้เกิดการปรับเปลี่ยนค่าน้ำหนักประสาทหรือแม้กระทั่งเงื่อนไขในการฝึกฝนของเครือข่าย

Artificial Neural Networks (ANN)?

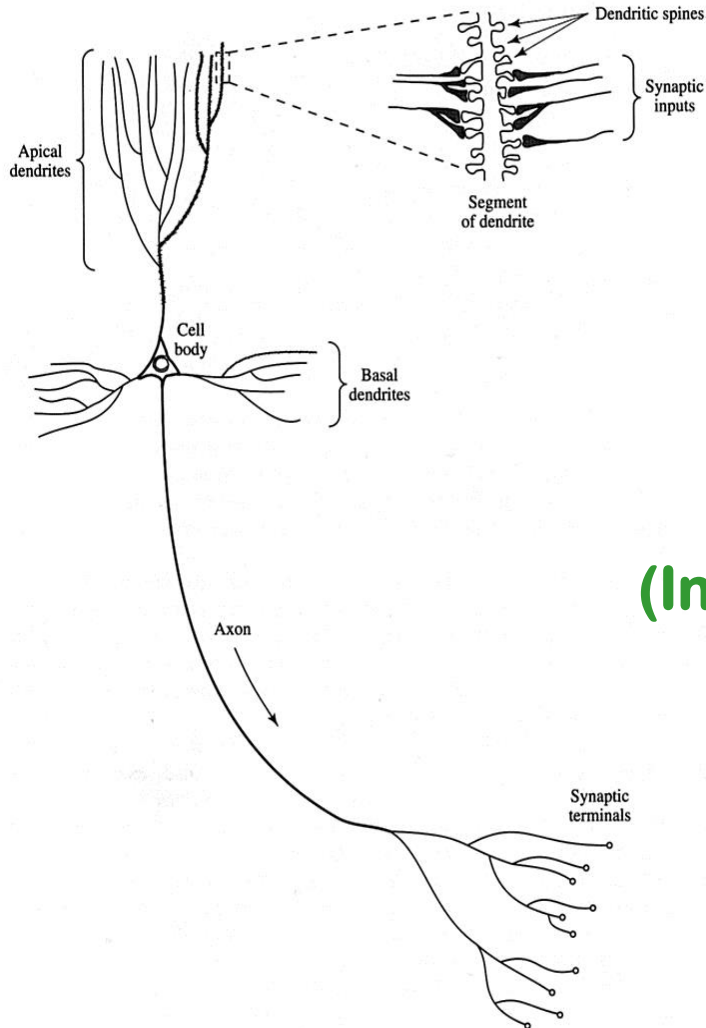
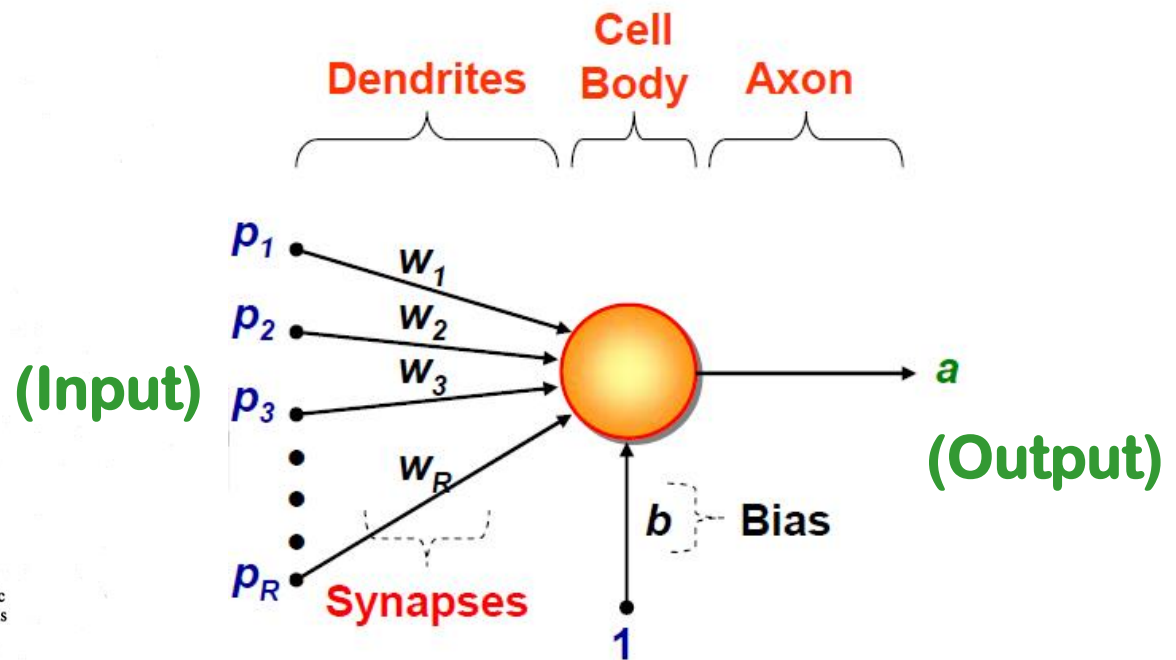


FIGURE 1.1 The pyramidal cell.

Component of Simple Neuron



Neural Network VS. General Computer

- เครือข่ายนิวรอลเน้นที่โครงสร้างเป็นหลัก ใช้หน่วยประมวลผลง่ายๆ จำนวนมากมายต่อกันขณะที่คอมพิวเตอร์ทั่วไปใช้หน่วยประมวลผลความสามารถสูง แต่มีจำนวนไม่มาก
- การโปรแกรมของคอมพิวเตอร์โดยทั่วไป ใช้ชุดคำสั่งเป็นลำดับขั้นตอน แต่เครือข่ายนิวรอลจะเรียนรู้โดยการฝึกฝนจาก ชุดข้อมูลสำหรับฝึกหัด (Training set)
- เครือข่ายนิวรอลจดจำได้โดยการปรับค่า weight ของ connections ที่ทำให้เครือข่ายมีข้อผิดพลาดจากการฝึกหัด (training error) ต่ำที่สุด
- การปรับ weight จะค่อยๆปรับทีละน้อยในการฝึกแต่ละครั้ง เมื่อฝึกบ่อยๆ ค่าความผิดพลาดก็จะลดลงเรื่อยๆ
- ปัจจุบันโปรแกรมเครือข่ายนิวรอลมักจะใช้การจำลองบนคอมพิวเตอร์แทนส่วนที่เป็นเครือข่ายฮาร์ดแวร์ซับซ้อน โดยใช้ซอฟต์แวร์เป็นหลัก ส่วนฮาร์ดแวร์ที่เลียนแบบเครือข่ายนิวรอลโดยตรงมีน้อยมาก เนื่องจากความยากลำบากในการสร้าง

History of Neural Networks (1)

- Warren S. McCulloch และ Walter Pitts [McCulloch and Pitts, 1943] แบบจำลองนิรอนของ McCulloch-Pitts เป็นจุดเริ่มต้นของเครือข่ายประสาทเทียม ยังไม่มีการนำเสนอวิธีการฝึกสอนใดๆ แต่นิรอนดังกล่าวสามารถทำงานทางลอจิกบางอย่างได้
- Donald Hebb [Hebb, 1949] จุดเริ่มต้นของการนำเสนอกระบวนการเรียนรู้ (learning process) ถือว่าเป็นครั้งแรกที่มีการนำเสนอการปรับค่าน้ำหนักประสาท
- John von Neumann [von Neumann, 1958] เล็งเห็นถึงศักยภาพความสัมพันธ์ระหว่างดิจิทัลคอมพิวเตอร์และการทำงานของสมอง เช่น หน่วยความจำของคอมพิวเตอร์เป็นต้น
- Frank Rosenblatt [Rosenblatt, 1958, 1960, 1962] ผู้เริ่มต้นแนวคิดของเพอร์เซ็ปตรอน (perceptron) งานดังกล่าวถือว่าเป็นครั้งแรกที่ได้มีการนิยามเครือข่ายประสาทเทียมได้อย่างชัดเจน
- Bernard Widrow และ Marcian Hoff [Widrow and M. E. Hoff, 1960, M. E. Hoff, 1962, Widrow, 1962, 1987] นำเสนอ ADALINE (ADaptive LINear Element) ซึ่งทำการฝึกฝนโดยอัลกอริทึมกำลังสองน้อยที่สุดหรือ LMS (least-mean-square) และขยายผลเป็น MADALINES (Many ADALINES) ในเวลาต่อมา

History of Neural Networks (2)

- Marvin Minsky และ Seymour Papert [Minsky and Papert, 1969] ได้นำเสนอข้อจำกัดของเครือข่ายประสาทเทียมแบบชั้นเดียว ทำให้เกิดการลังเลในทิศทางการพัฒนาของเครือข่ายประสาทเทียม
- Teuvo Kohonen และ James Anderson [Kohonen, 1972, Anderson, 1972] การจดจำด้วยเนื้อหาสัมพันธ์ (content-addressable associative memory)
- Paul Werbos [Werbos, 1974] ผู้เริ่มต้นแนวคิดของอัลกอริทึมแบบแพร่กลับ (backpropagation)
- John Hopfield [Hopfield, 1982, 1984] เข้าสู่ยุคใหม่ Hopfield แสดงให้เห็นถึงการทำงานที่ซับซ้อนและครอบคลุมของเครือข่ายป้อนกลับ (recurrent network)
- Teuvo Kohonen [Kohonen, 1982, 1989, 1990] นำเสนอแผนผังคุณลักษณะการจัดการตัวเอง (self-organizing feature map) แบบ unsupervised learning
- Gail Carpenter และ Stephen Grossberg [Carpenter and Grossberg, 1987] พัฒนาเครือข่าย ART
- David Broomhead และ David Lowe [Broomhead and Lowe, 1988] นำเสนอ RBF

Types of Neural Networks

Categorizes by Architectures

- เครือข่ายไปข้างหน้า หรือ feed-forward network
- เครือข่ายป้อนกลับ หรือ recurrent network หรือ feedback network
(Ref. รูปที่ 8.1 อาทิตย์ ศรีแก้ว)

Categorizes by Learning

- การเรียนรู้แบบไม่มีผู้ฝึกสอน (unsupervised learning)
- การเรียนรู้แบบมีผู้ฝึกสอน (supervised learning)
- การเรียนรู้แบบเสริมความเข้มแข็ง (Reinforcement Learning)

Applications of ANN

- Aerospace
- Automotive
- Banking
- Credit Card Activity Checking
- Defense
- Electronics
- Entertainment
- Industrial
- Insurance
- Manufacturing
- Medical
- Oil & Gas
- Robotics
- Speech
- Securities
- Telecommunications
- Transportation
- Etc.,

Examples of Detail Applications

- การจดจำรูปแบบ (pattern recognition)
- การจับกลุ่ม (clustering/categorization)
- การประมาณค่าฟังก์ชัน (function approximation)
- การทำนาย (prediction/forecasting)
- การหาค่าเหมาะที่สุด (optimization)
- หน่วยความจำอ้างอิงด้วยเนื้อหา (content-addressable memory)
- ระบบควบคุม (control system)
- Etc.,

จุดเด่นของ ANN

- การเชื่อมต่อแบบขนานเป็นจำนวนมาก (massive parallel)
- ข้อมูลและการคำนวณแบบกระจาย (distributed data and computation)
- ความสามารถในการเรียนรู้ (learning)
- ความสามารถในการทำให้เป็นทั่วไป (generalization)
- การปรับตัว (adaptation)
- การประมวลผลข้อมูลเชิงเนื้อหา (content-based processing)
- การทนต่อความพ้อง (fault tolerance)

Artificial Neural Networks (ANN)

Neural Network Architectures

Institute of Engineering
School of Computer Engineering
Suranaree University of Technology

Typically Steps in Supervised NN

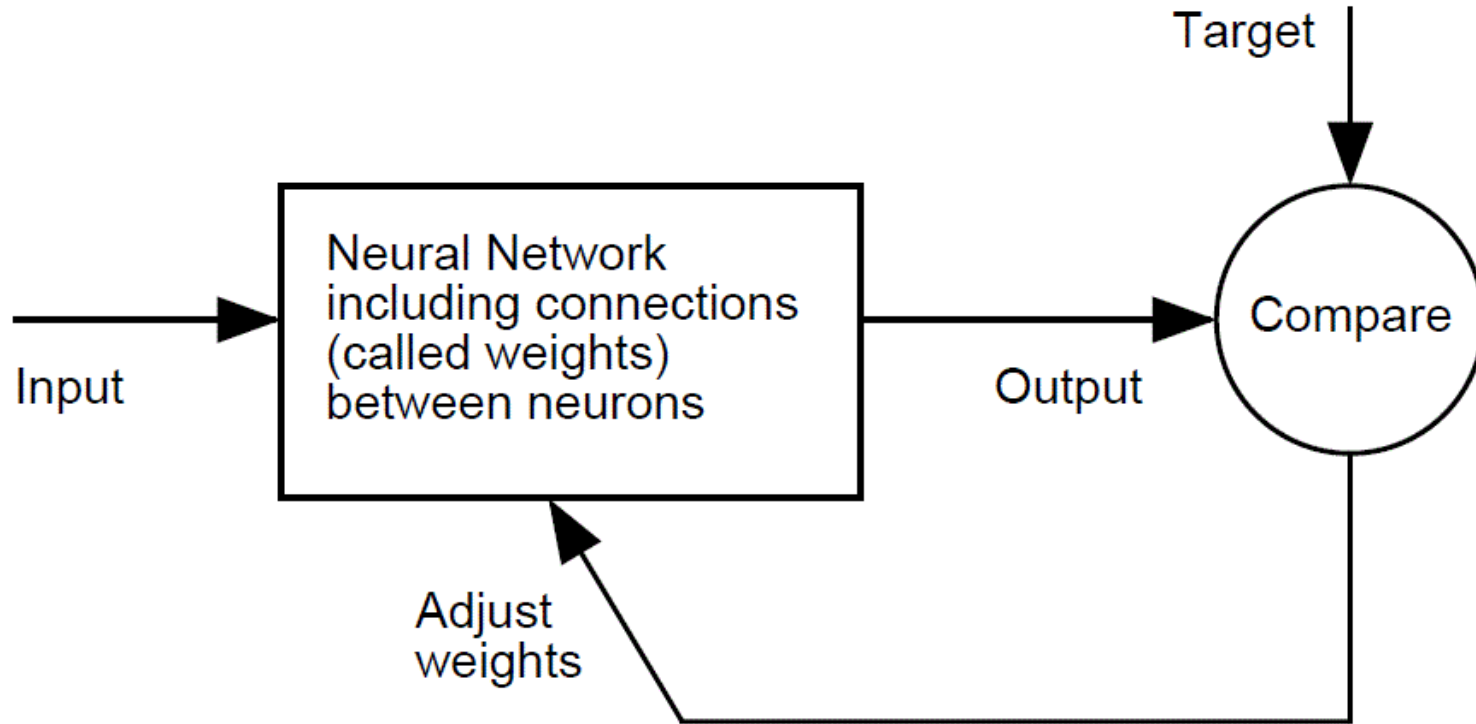
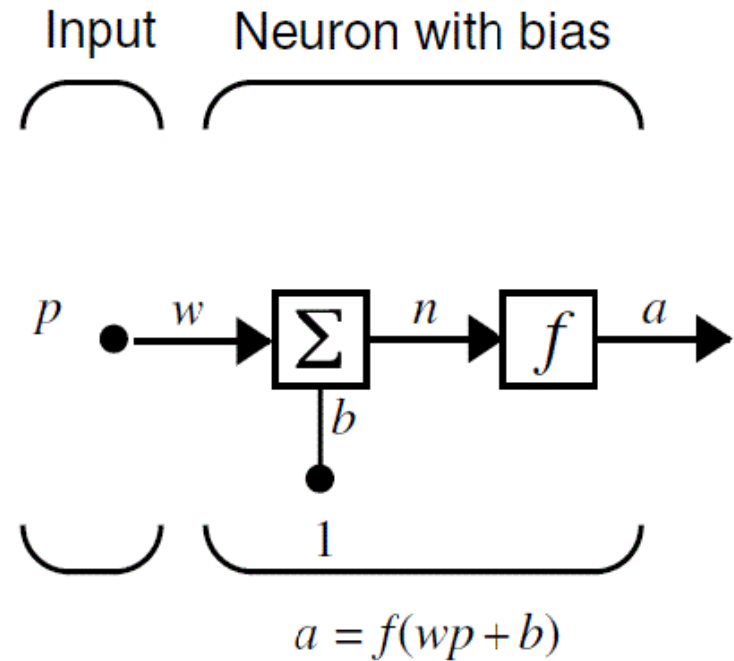
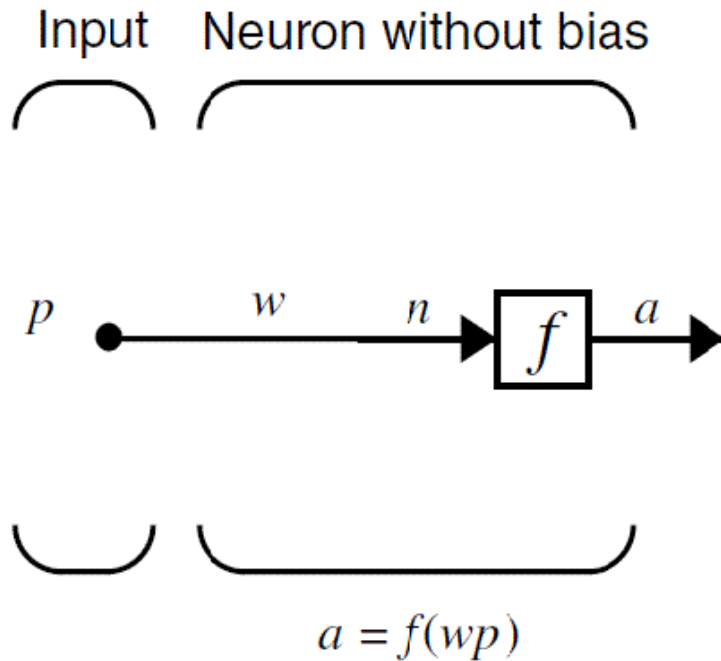


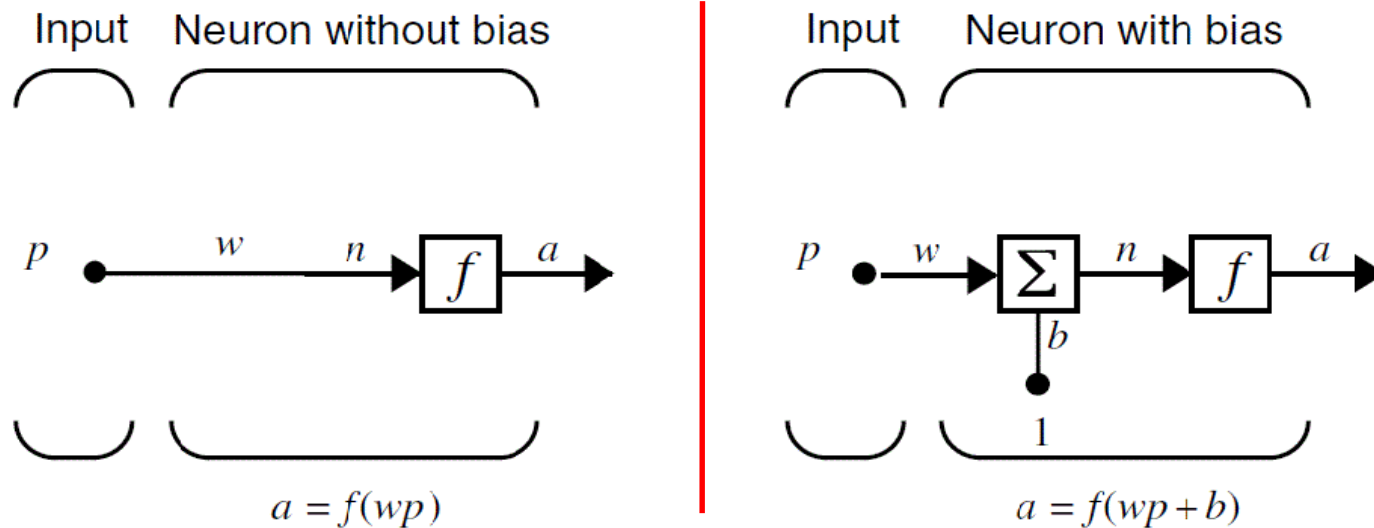
Fig. Ref. NN Toolbox User's Guide Version 4

Simple Neuron with A Single Input



- Single neuron with single input
(single scalar input: each input has 1 element or 1 component or 1 dimension)

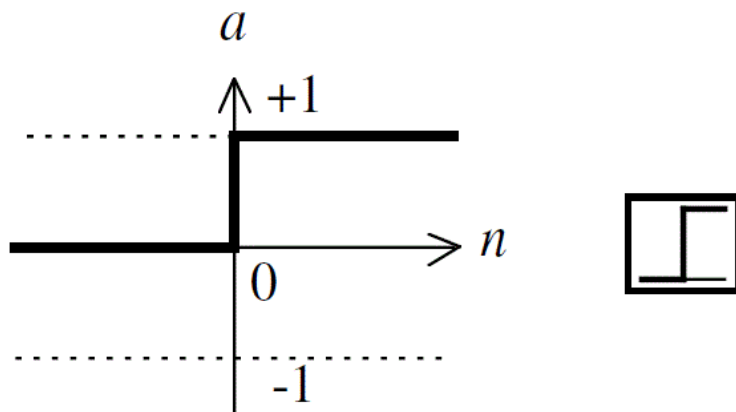
Simple Neuron with A Single Input



➤ Single neuron with single input parameters

- p : an input
- w : weight
- wp : weighted input
- b : bias (or offset)
- n : net input
- f : transfer function (activation function)
- a : an output (sometimes we may use y)

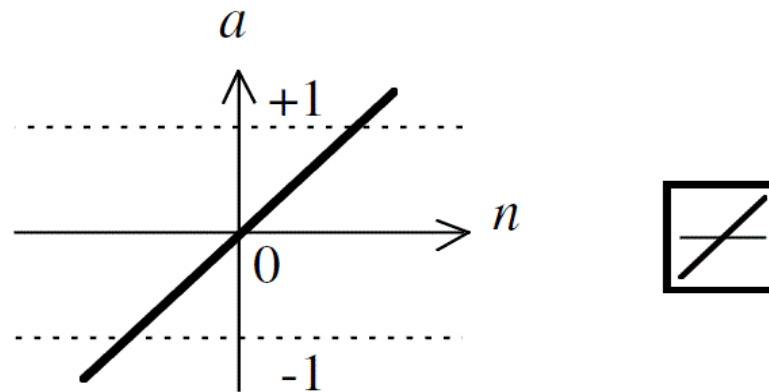
Transfer Functions



$$a = \text{hardlim}(n)$$

$$\begin{aligned} a &= 0, \text{ If } n < 0 \\ a &= 1, \text{ If } n \geq 0 \end{aligned}$$

Hard-Limit transfer function

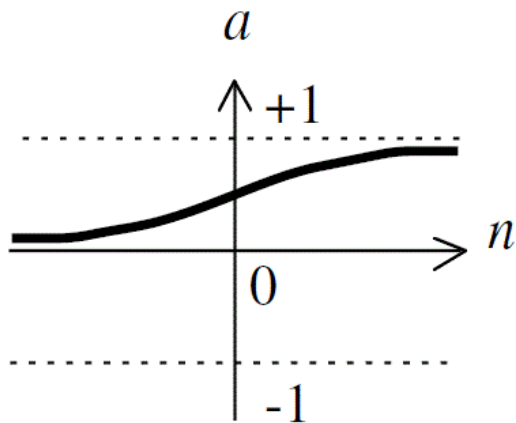


$$a = \text{purelin}(n)$$

$$a = n$$

Linear transfer function

Transfer Functions



$$a = \text{logsig}(n)$$

$$a = \frac{1}{1 + e^{-n}}$$

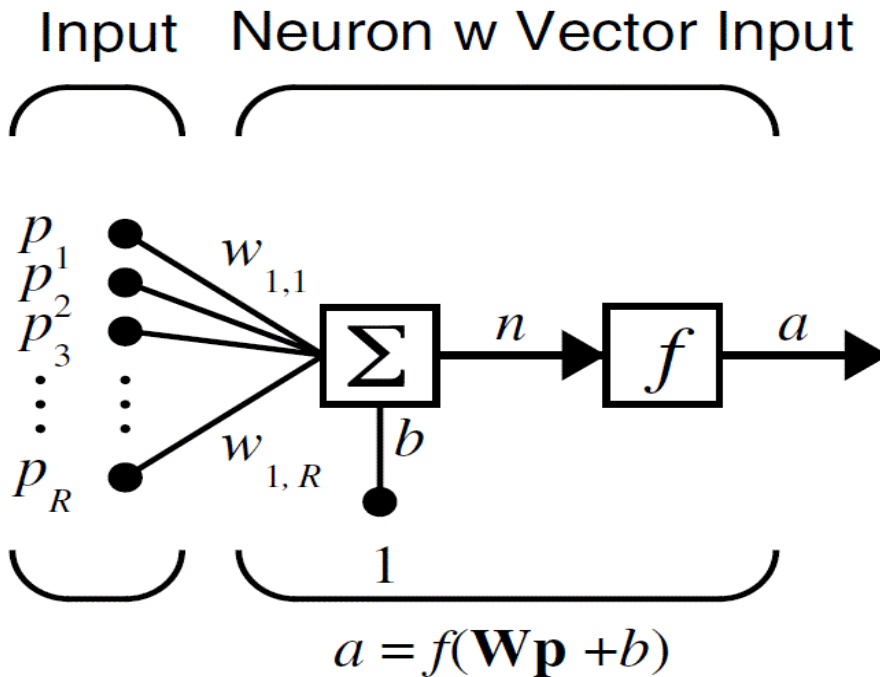
Log-Sigmoid transfer function

➤ Other Transfer Functions

Read : ตาราง 8.5 Student Ref.
(ปัญญาเชิงคำนวณ, ผศ.ดร. อาทิตย์ ศรีแก้ว)

- Symmetrical hard limit
- Positive linear
- Hyperbolic tangent sigmoid
- Competitive
- Radial basis function

Single (Individual) Neuron with Multiple Input



Where...

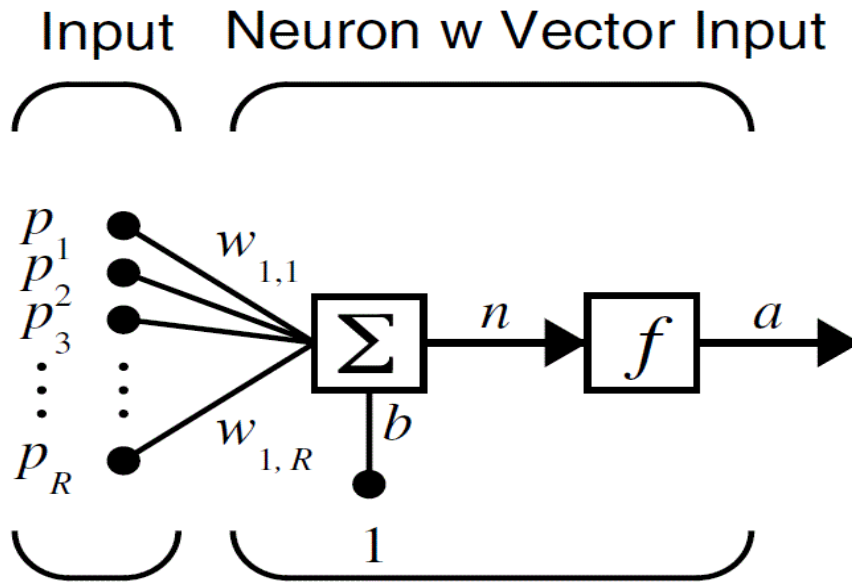
R = number of elements in
input vector
(each input has R dimension)

\mathbf{Wp} : the dot product of the (single row) matrix \mathbf{W} and the vector \mathbf{p}

$$\mathbf{p} = p_1, p_2, \dots, p_R$$

$$\mathbf{W} = w_{1,1}, w_{1,2}, \dots, w_{1,R}$$

Single (Individual) Neuron with Multiple Input



Where...

R = number of elements in
input vector
(each input has R dimension)

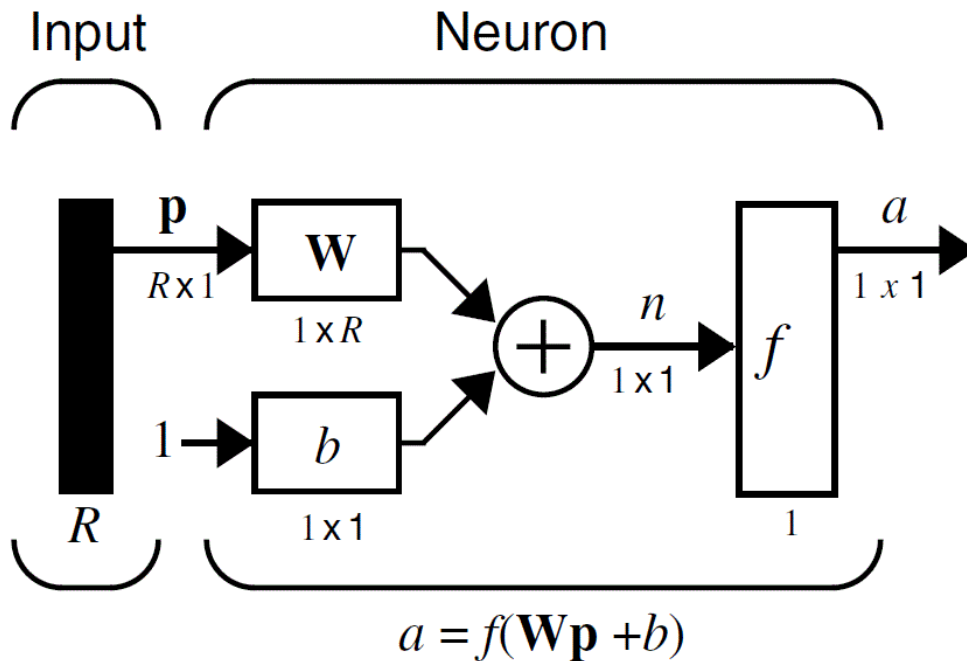
$$a = f(\mathbf{W}\mathbf{p} + b)$$

$$a = f(n)$$

$$n = w_{1,1}p_1 + w_{1,2}p_2 + \dots + w_{1,R}p_R + b$$

$$\mathbf{n} = \mathbf{W} * \mathbf{p} + \mathbf{b}$$

Single (Individual) Neuron with Multiple Input (Abbreviated Notation)



Where...

R = number of elements
in input vector
(each input has R
dimension)

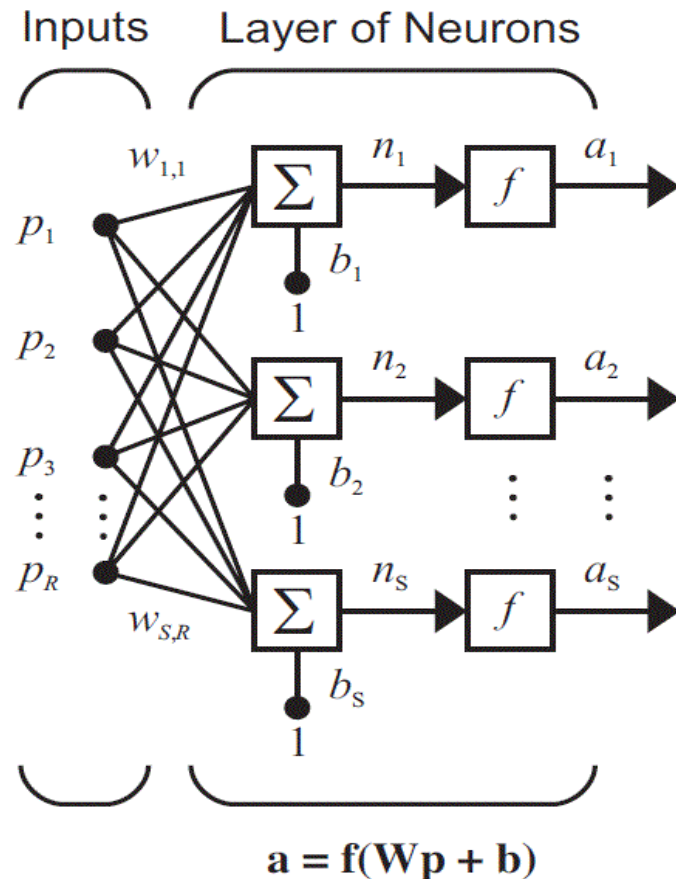
➤ Transfer function may
specify by symbols:

hardlim

purelin

logsig

A Layer of Neurons



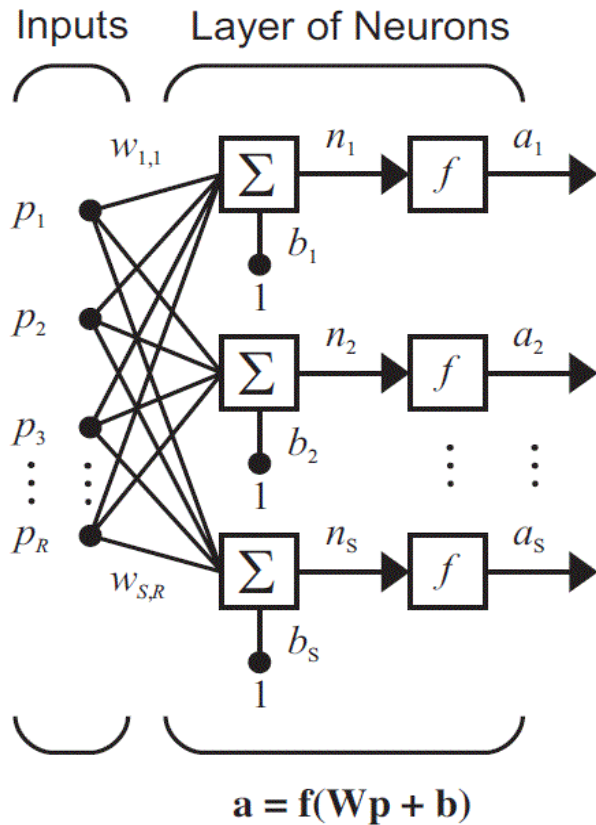
Where

R = number of
elements in
input vector

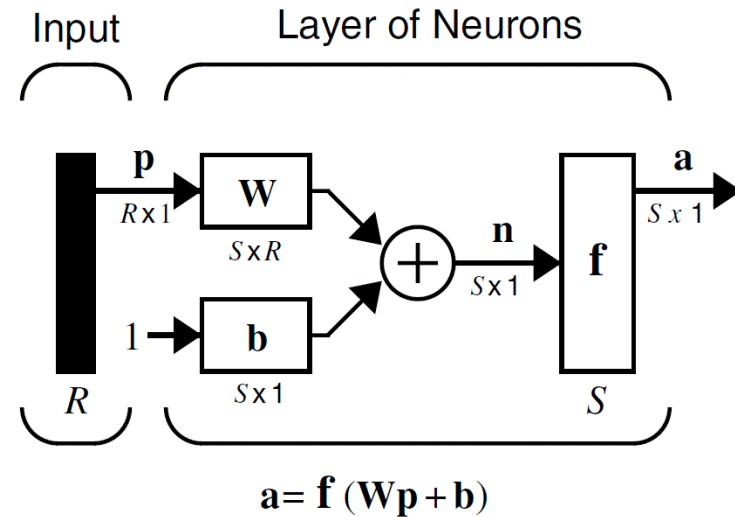
S = number of
neurons in layer

A one-layer network with R input elements and S neurons

A Layer of Neurons (Abbreviated Notation)



R = number of input elements
 S = number of neurons in layer



Where...

$$\mathbf{p} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_R \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_S \end{bmatrix} \quad \mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,R} \\ w_{2,1} & w_{2,2} & \dots & w_{2,R} \\ \vdots & \vdots & \ddots & \vdots \\ w_{S,1} & w_{S,2} & \dots & w_{S,R} \end{bmatrix}$$

Multiple Layers of Neurons

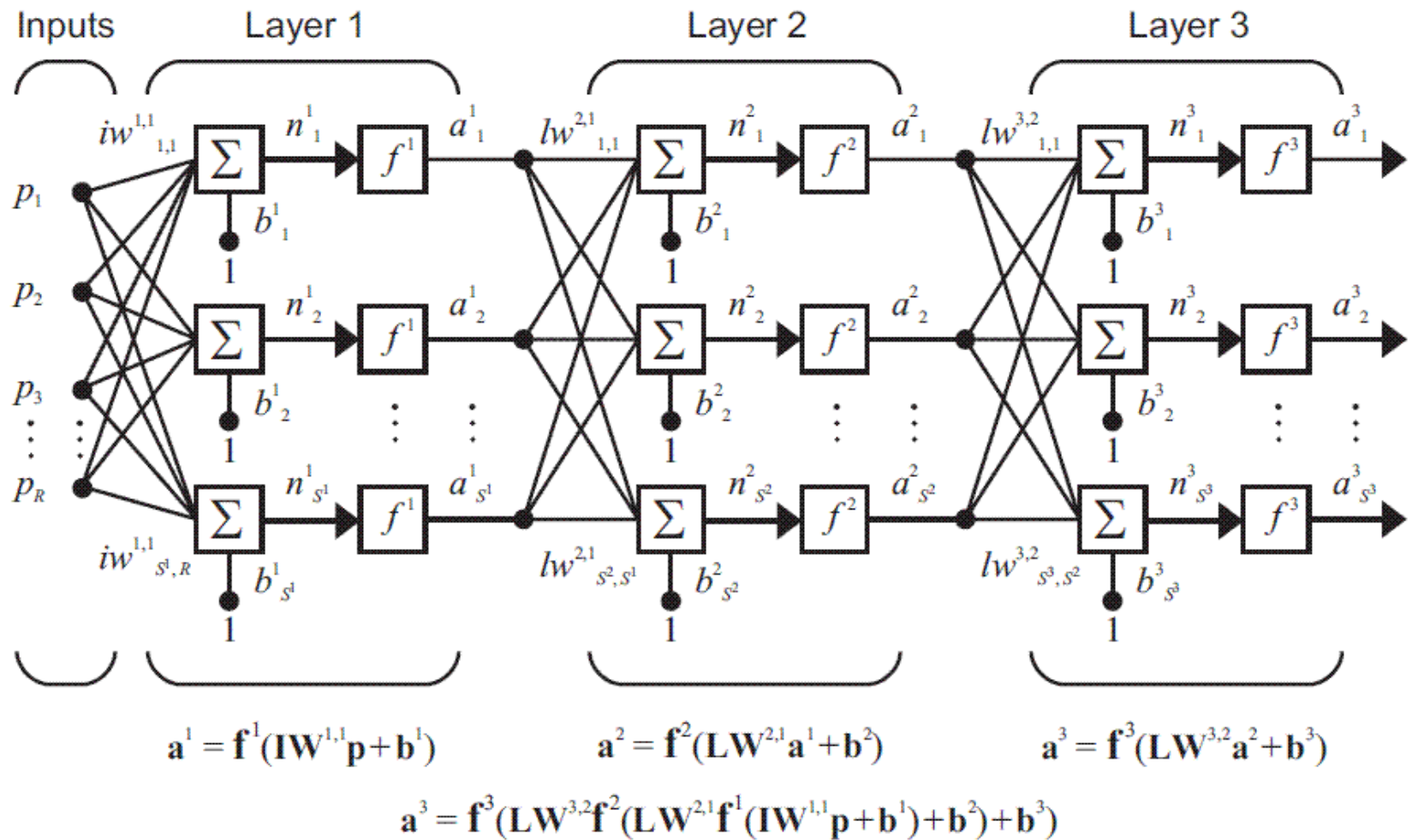


Fig. Ref. NN Toolbox User's Guide Version 6

Multiple Layers of Neurons (Abbreviated Notation)

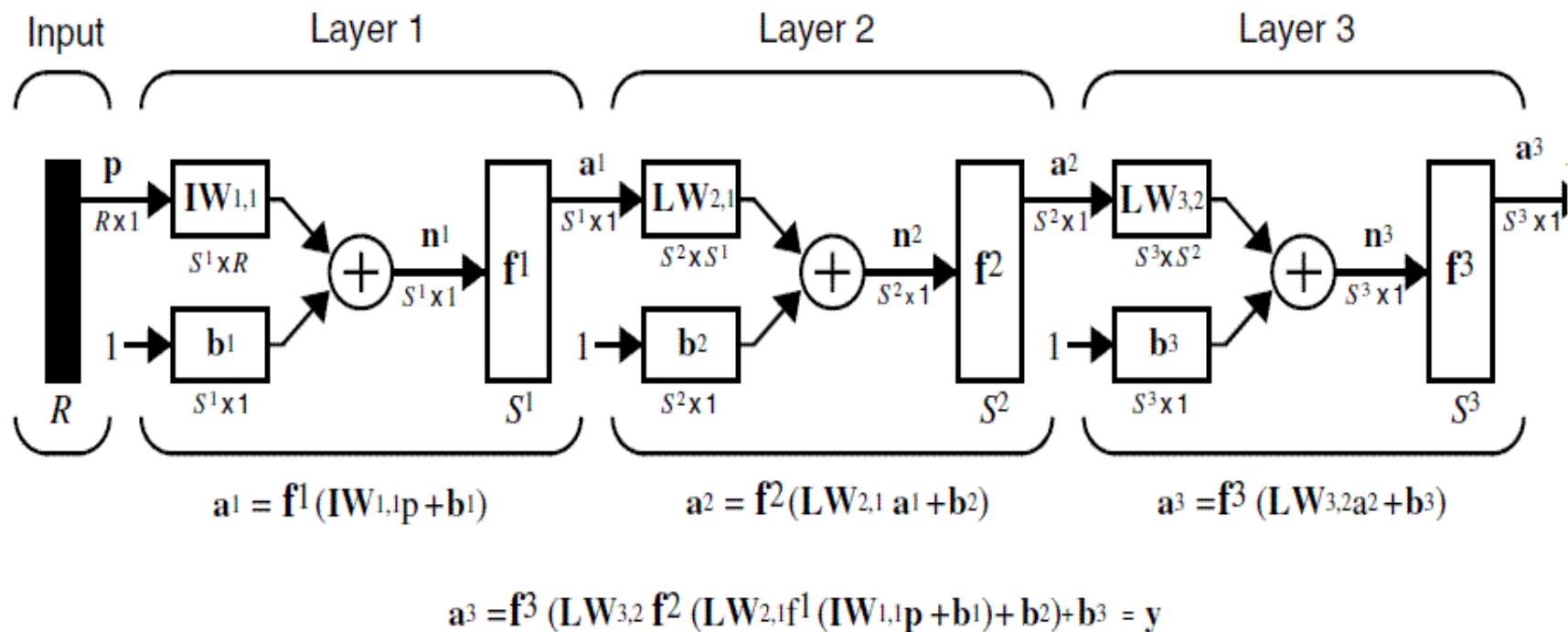
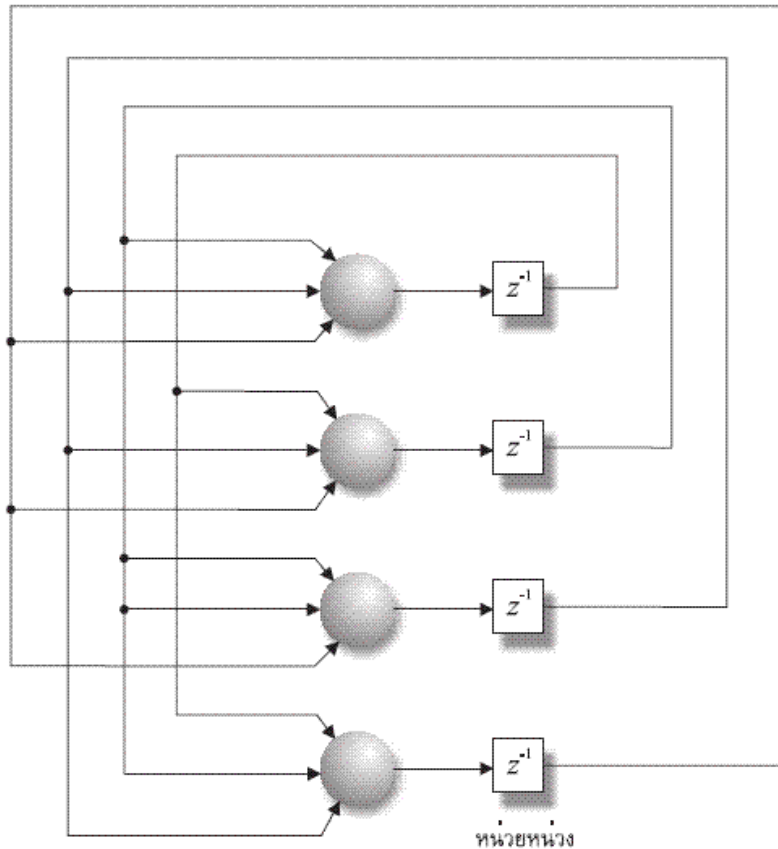
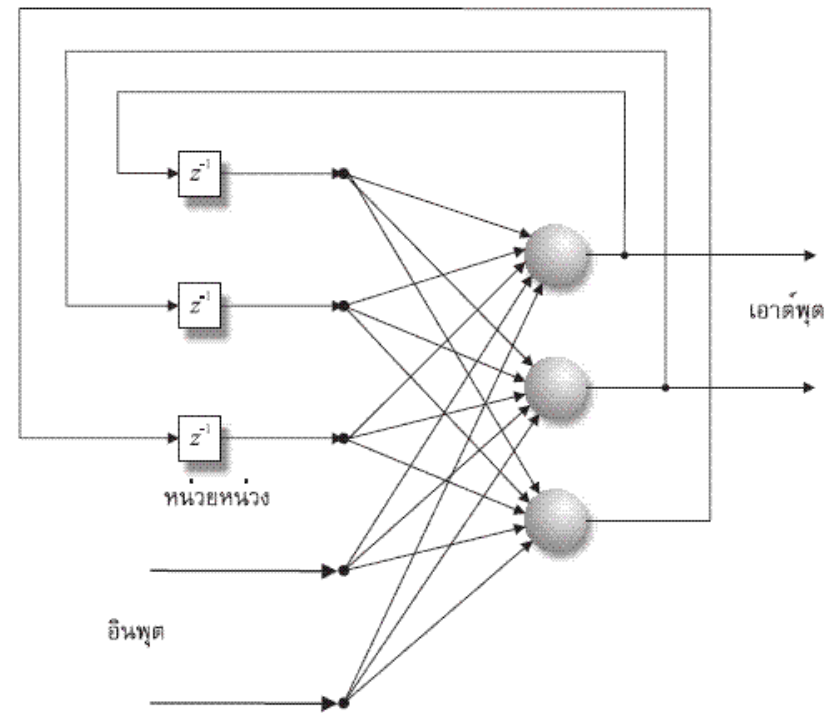


Fig. Ref. NN Toolbox User's Guide Version 6

Recurrent (Feedback) Network



รูปที่ 8.19: เครือข่ายป้อนกลับที่ไม่มีการป้อนกลับให้ตนเองและไม่มีนิวรอนซ่อนเร้น

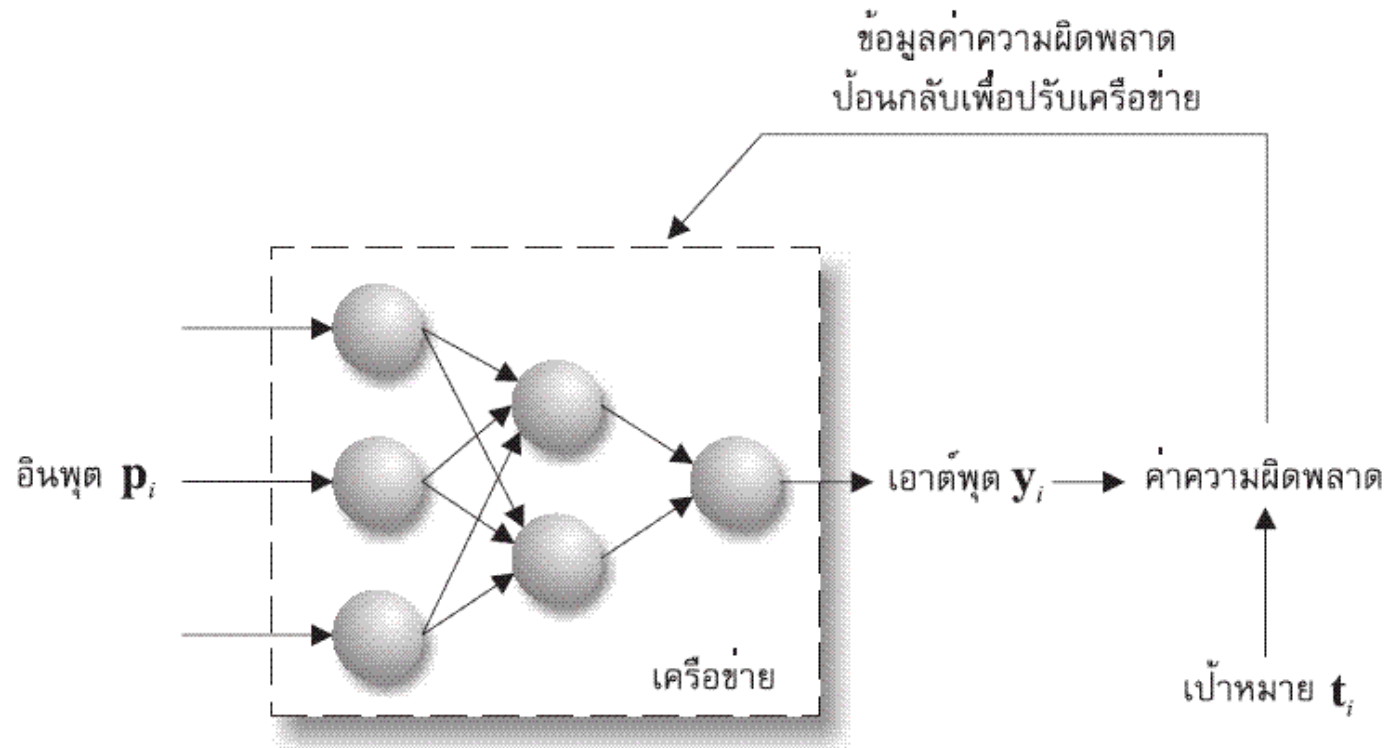


รูปที่ 8.20: เครือข่ายป้อนกลับที่มีนิวรอนซ่อนเร้น

Fig. Ref. ปัญญาเชิงคำนวณ, ผศ.ดร. อาทิตย ศรีแก้ว

Learning Rules

- **Supervised Learning** มีการนำเสนอกลุ่มข้อมูลตัวอย่าง (examples หรือ training set) ให้กับเครือข่ายในรูปของคู่อินพุตพร้อมกับเป้าหมาย (target)



รูปที่ 8.22: ขั้นตอนการเรียนรู้แบบมีผู้ฝึกสอน

Fig. Ref. ปัญญาเชิงคำนวณ, ผศ.ดร. อาทิตย์ ศรีแก้ว

Learning Rules

- **Unsupervised Learning** เครือข่ายจะทำการปรับค่าน้ำหนักประสาทและไบอัส โดยใช้ข้อมูลจากอินพุตเท่านั้น นั่นคือไม่มีเป้าหมายให้ใช้ในการเปรียบเทียบ

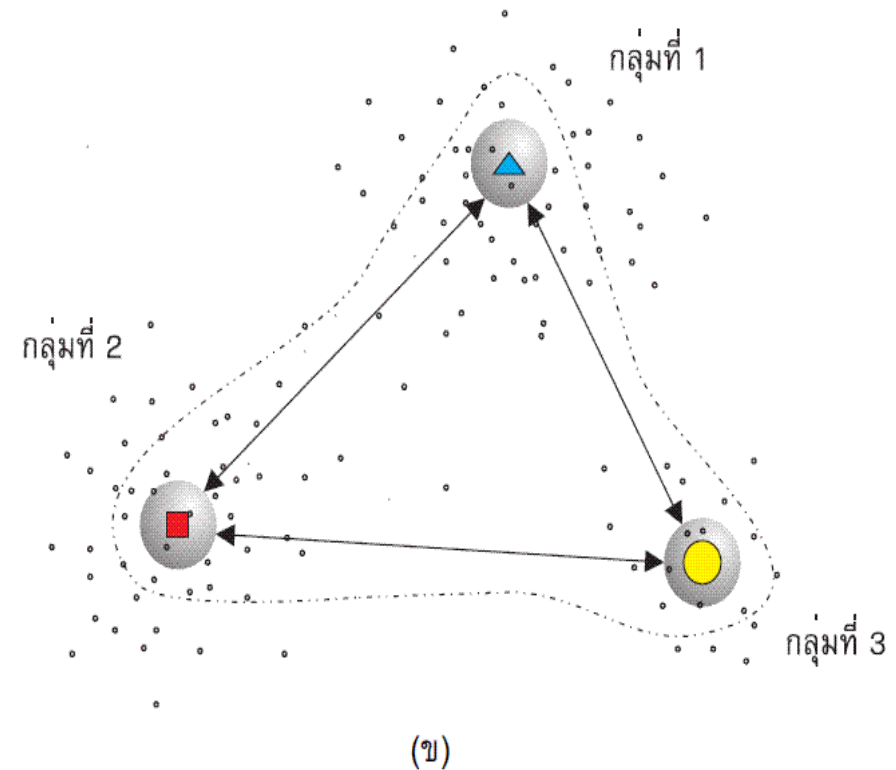
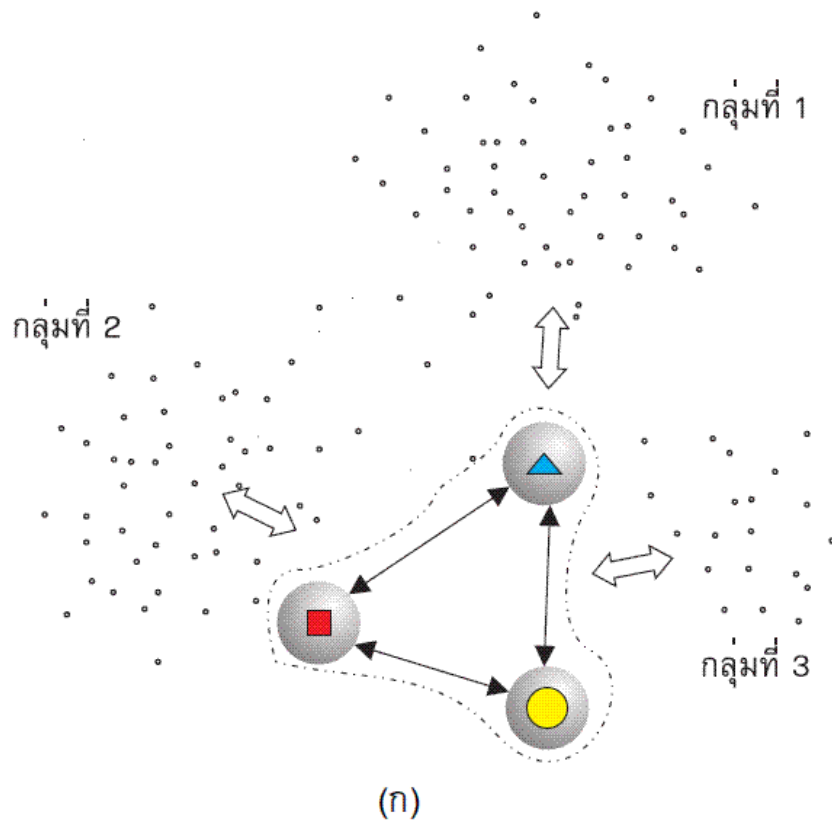
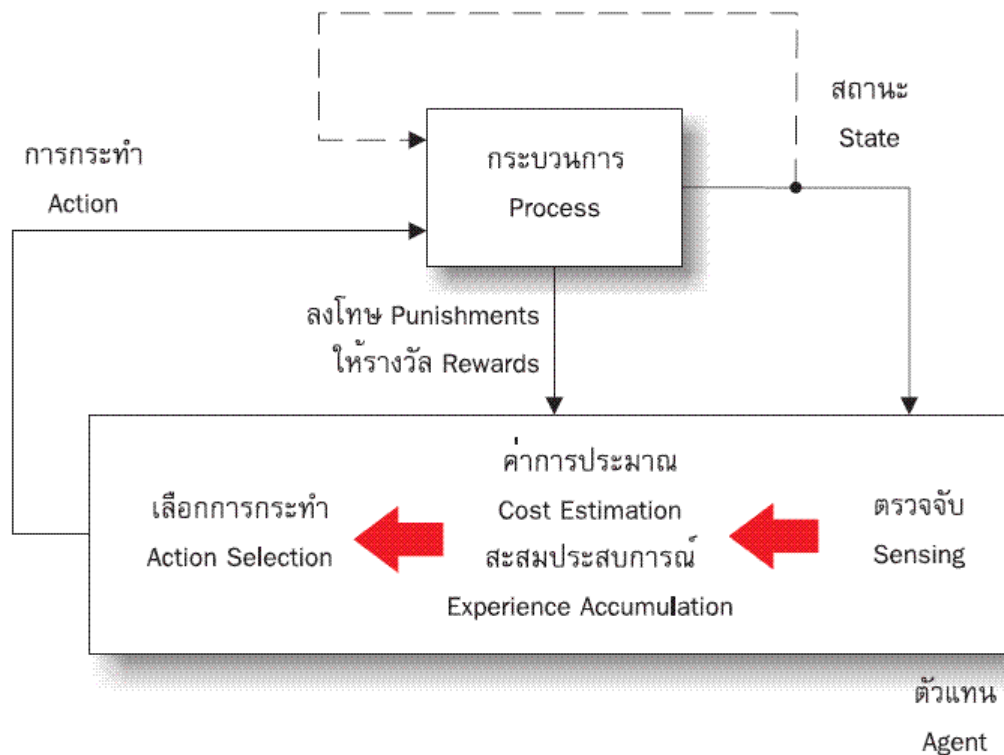


Fig. Ref. ปัญญาเชิงคำนวณ, ผศ.ดร. อาทิตย์ ศรีแก้ว

Learning Rules

➤ **Reinforcement Learning** จะให้คะแนนประสิทธิภาพของเครือข่ายในรูปการลงโทษ (เมื่อเครือข่ายทำงานผิดเป้าหมาย) หรือให้รางวัล (เมื่อเครือข่ายทำงานถูกต้องตามเป้าหมาย)



รูปที่ 8.24: แผนผังการทำงานของ การเรียนรู้แบบเสริมความเข้มแข็ง

Fig. Ref. ปัญญาเชิงคำนวณ, ผศ.ดร. อาทิตย์ ศรีแก้ว