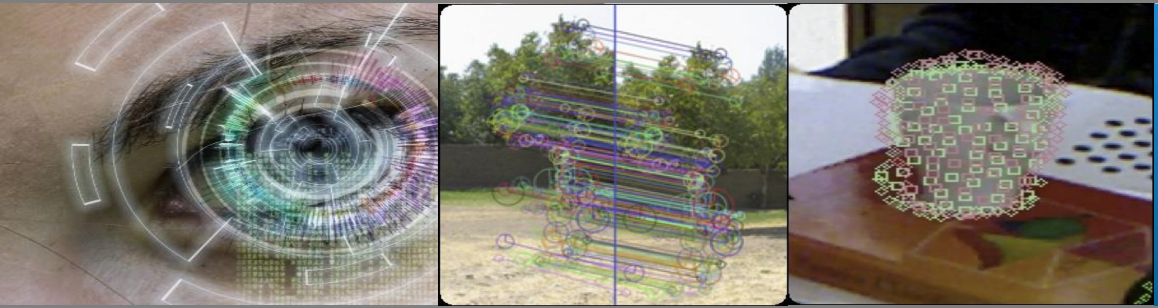
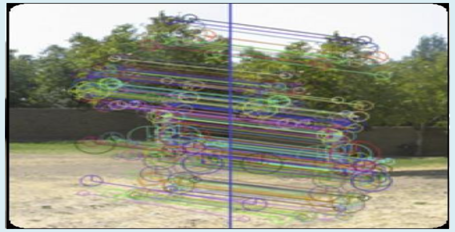
Computer Vision

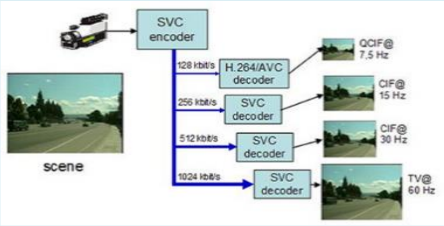
Deep Learning: Artificial Neural Networks (ANN)(2)
- LMS Algorithm & Multilayer Perceptron & Back Propagation



2023 Fall

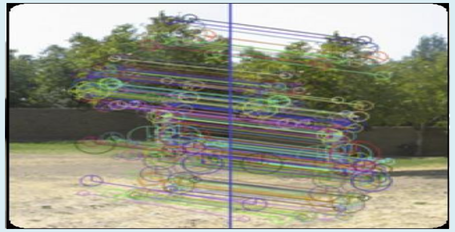
Prof. Byung-Gyu Kim
Intelligent Vision Processing Lab. (IVPL)
http://ivpl.sookmyung.ac.kr
Dept. of IT Engineering, Sookmyung Women's University
E-mail: bg.kim@ivpl.sookmyung.ac.kr

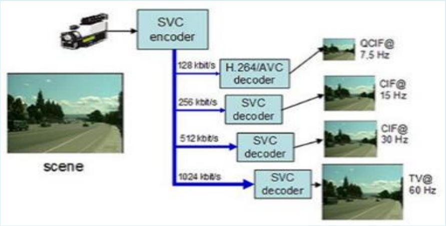




Contents

- Multilayer Perceptron (MLP)
- Back-Propagation Algorithm (BPA)
- Regression Analysis



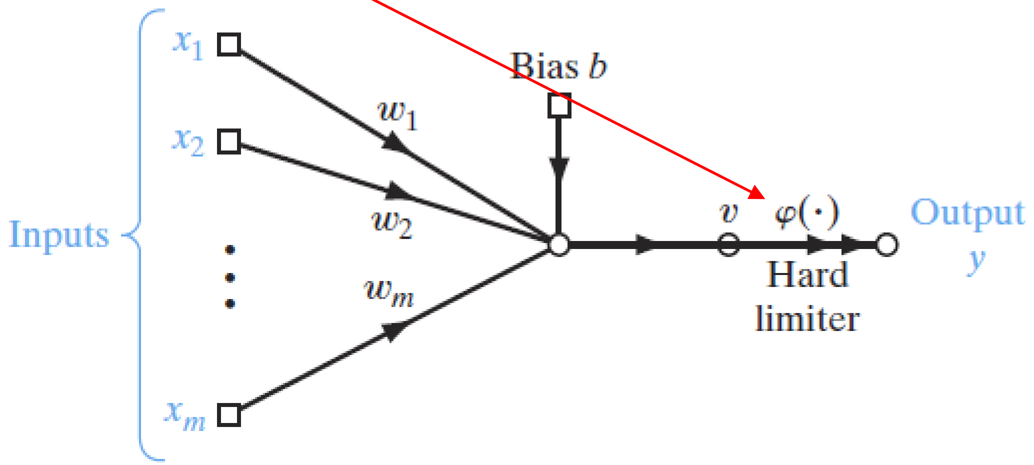


Contents

- **Multilayer Perceptron (MLP)**
- Back-Propagation Algorithm (BPA)
- Regression Analysis

- ***Neural modeling*** consists of a **linear combiner** followed by **a hard limiter** (performing the **signum function**)
  - the neuron produces an output equal to 1 if the hard limiter input is positive, and -1 if it is negative.

FIGURE 1.1 Signal-flow graph of the perceptron.



the hard limiter input, or
induced local field, of the neuron: $v = \sum_{i=1}^{m} w_i x_i + b$

Multilayer Perceptron (MLP) (1)

❖ What is **MLP**?

- The multilayer perceptron, which **stands for a neural network** with **one or more hidden layers**

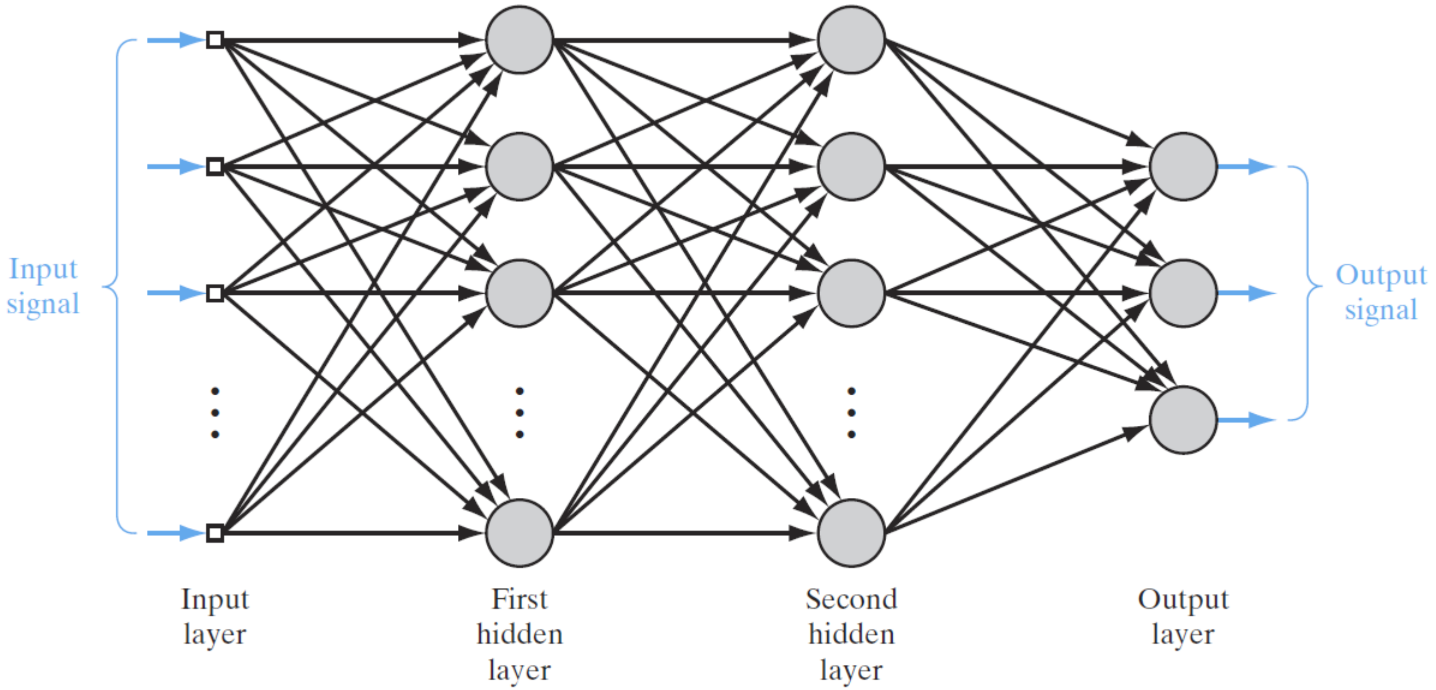


FIGURE 4.1 Architectural graph of a multilayer perceptron with two hidden layers. [*fully connected*]

Multilayer Perceptron (MLP) (2)

- Training: **two phases**
  - In the ***forward phase***,
    - the synaptic weights of the network are fixed and the input signal is propagated through the network, layer by layer, until it reaches the output.
  - In the ***backward phase***
    - an **error signal** is produced by comparing the output of the network with a desired response. The resulting error signal is propagated through the network, again layer by layer, but this time the propagation is performed in the backward direction.

"Back propagation algorithm"

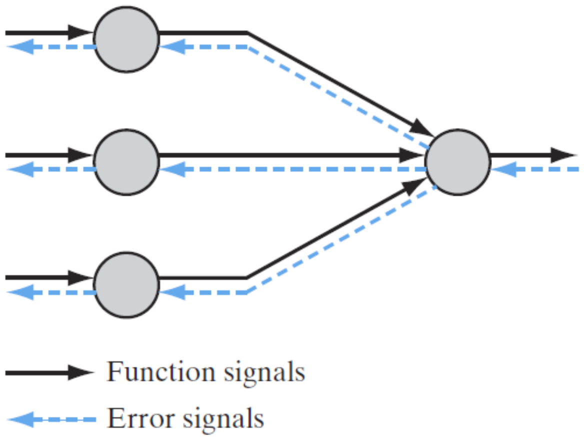
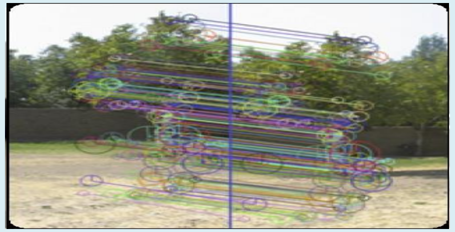
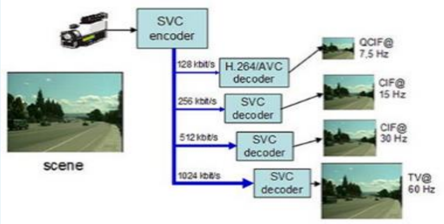


FIGURE 4.2 Illustration of the directions of two basic signal flows in a multilayer perceptron: forward propagation of function signals and back propagation of error signals.





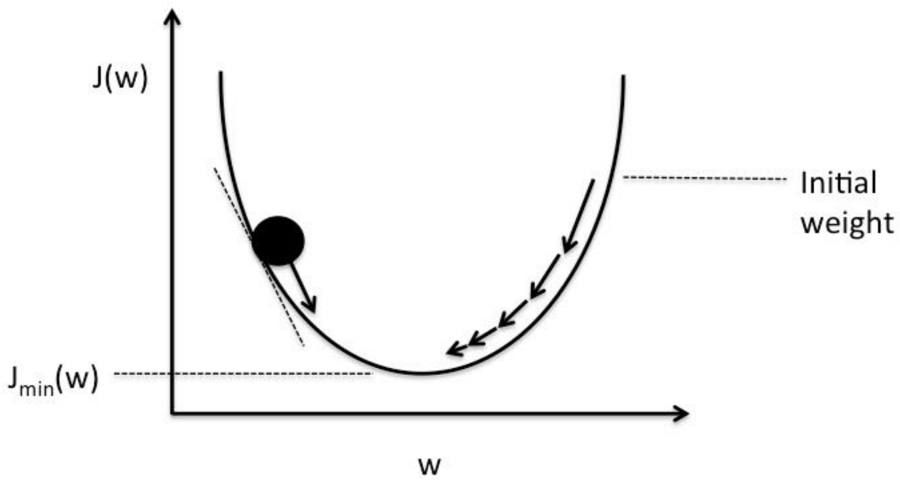
Contents

- Multilayer Perceptron (MLP)
- **Back-Propagation Algorithm (BPA)**
- Regression Analysis

Back-Propagation Algorithm (BPA) (1)

- The popularity of on-line learning for the supervised training of multilayer perceptrons
- ***How to correct amount of error term?***
  - ***Steepest Decent Method***

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{1}{2}\mu[-\nabla J(n)]. \qquad (5)$$



Schematic of gradient descent.

❖ For neuron *j* being fed by a set of function signals

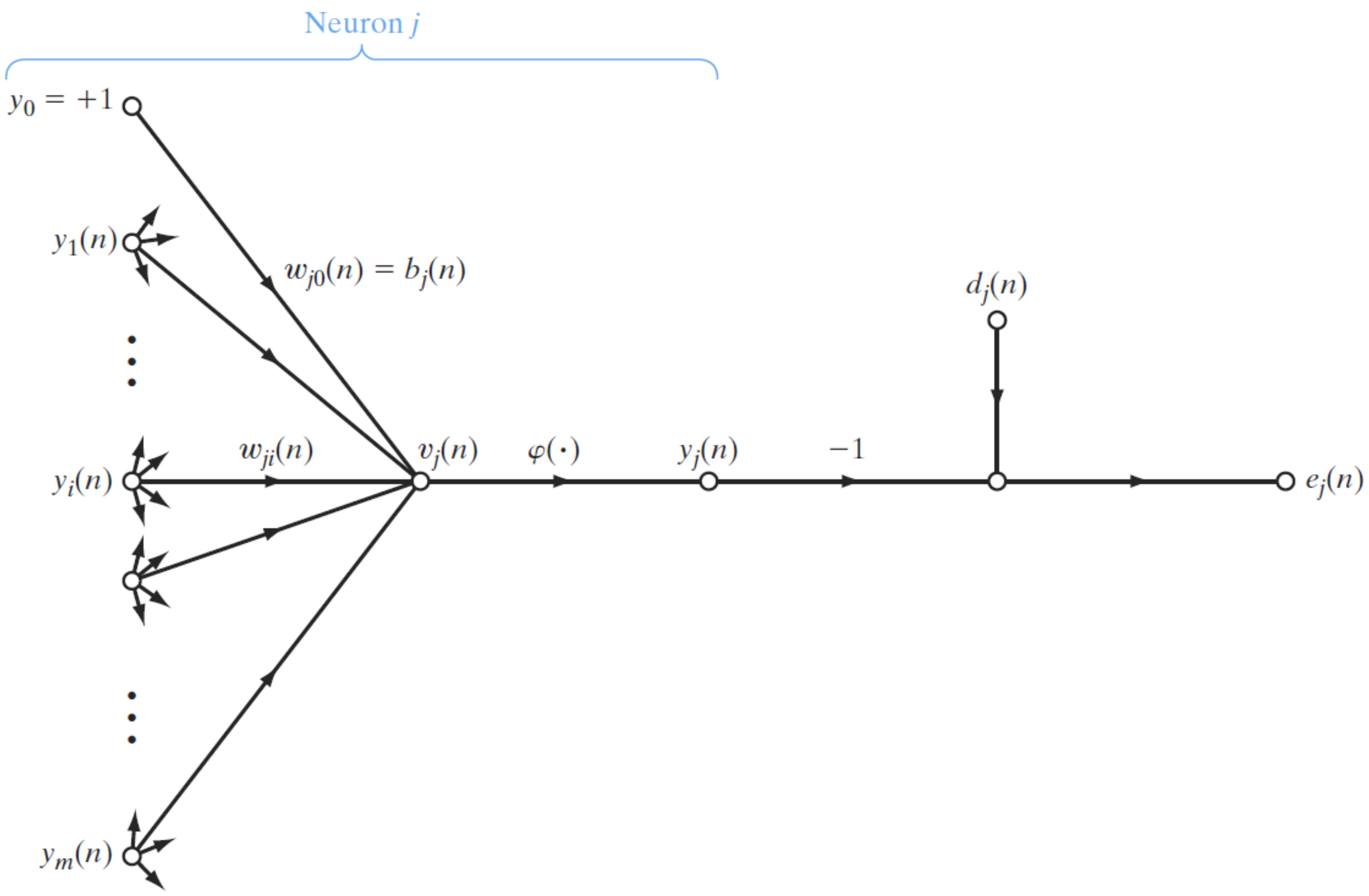


FIGURE 4.3 Signal-flow graph highlighting the details of output neuron *j*.

- **Error signal** $e_j(n)$

$$e_j(n) = d_j(n) - y_j(n)$$

where $d_j(n)$ is the jth element of the desired-response vector $\mathbf{d}(n)$.

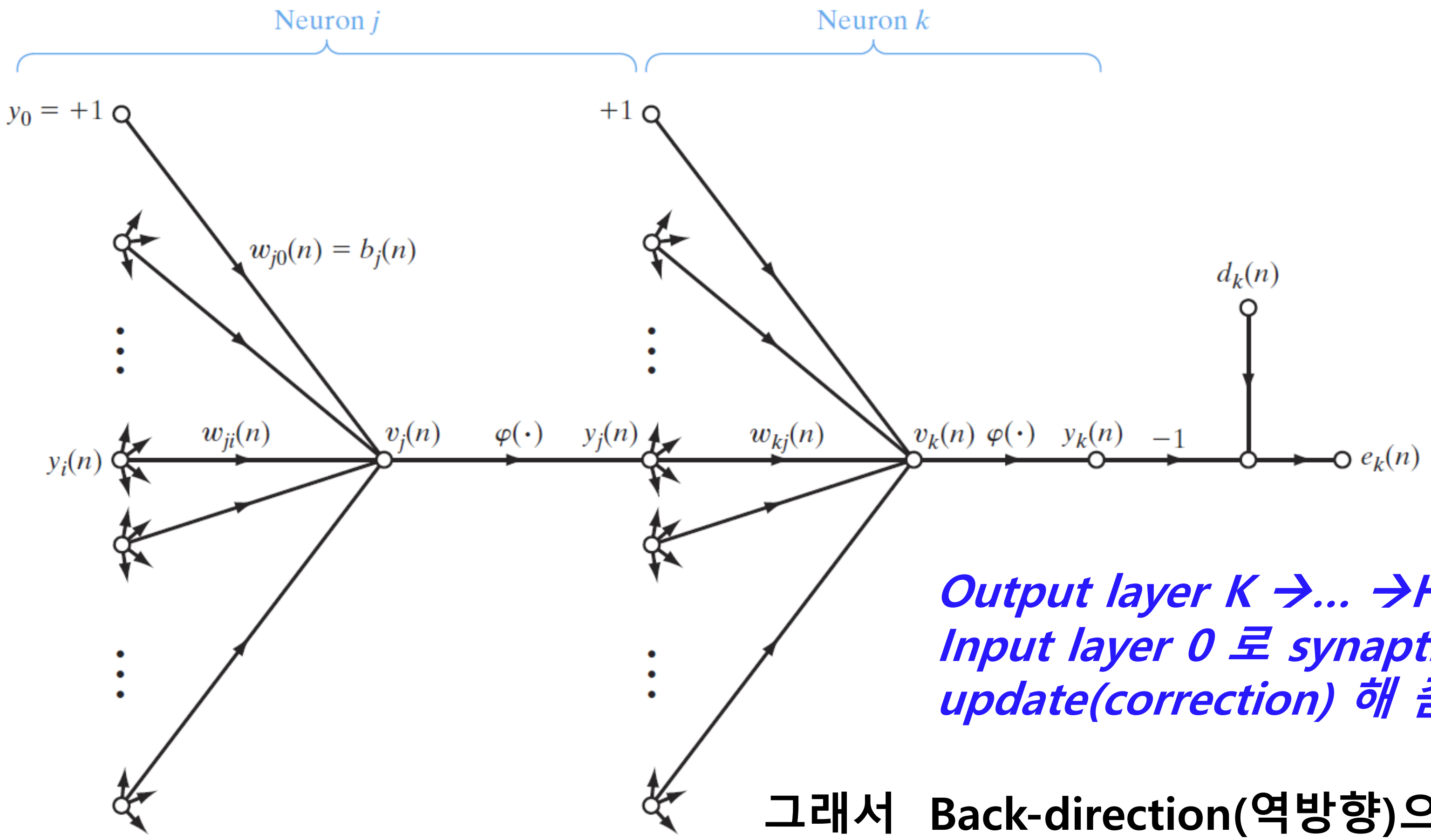
- ***instantaneous error energy*** of neuron *j*

$$\mathscr{E}_j(n) = \frac{1}{2} e_j^2(n)$$

Back-Propagation Algorithm (BPA) (3)

❖ Finally (at iteration *n*),

$$\mathbf{W}_{ji}(n+1) = \mathbf{W}_{ji}(n) + \Delta\mathbf{W}_{ji}(n)$$



Output layer K →… →Hidden layer j→…→ Input layer 0 로 synaptic weight를 순차적으로 update(correction) 해 줌

그래서 Back-direction(역방향)으로 전파해 준다는 의미가 BPA

FIGURE 4.4 Signal-flow graph highlighting the details of output neuron k connected to hidden neuron j.

Stopping Criteria

❖ How to define the convergence?

- In general, the back-propagation algorithm cannot be shown to converge, and there are no well-defined criteria for stopping its operation.

Let the weight vector **w*** denote a minimum, be it local or global. A necessary condition for **w*** to be a minimum is that the gradient vector **g(w)** (i.e., first-order partial derivative) of the error surface with respect to the weight vector **w** must be zero at **w = w***.

- *The back-propagation algorithm is considered to have converged when the Euclidean norm of the gradient vector reaches a sufficiently small gradient threshold.*

- *The back-propagation algorithm is considered to have converged when the absolute rate of change in the average squared error per epoch is sufficiently small.*

PERCEPTRON-error correction learning rule

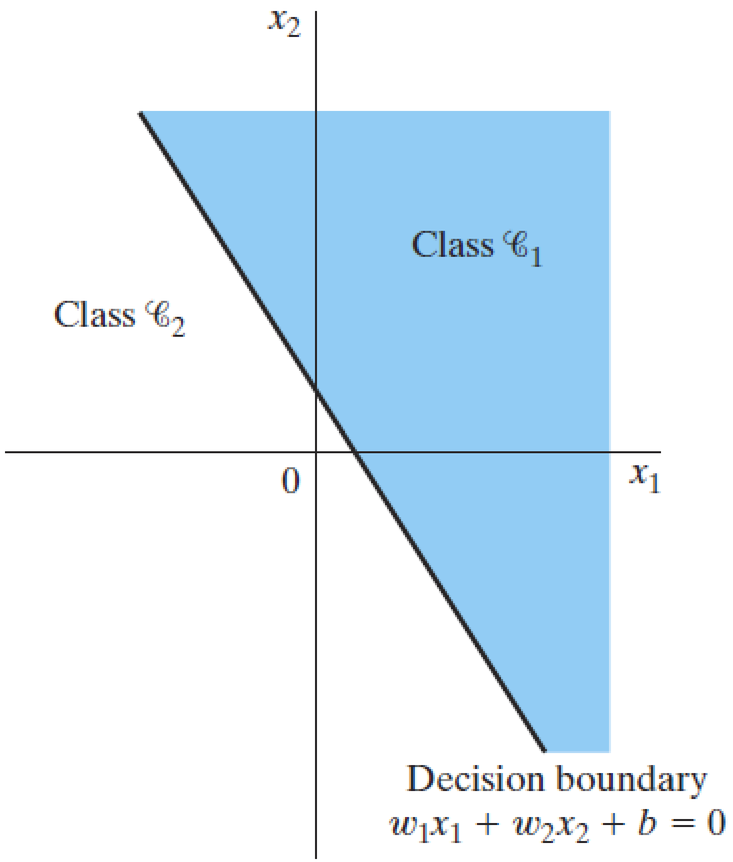


FIGURE 1.2 Illustration of the hyperplane (in this example, a straight line) as decision boundary for a two-dimensional, two-class pattern-classification problem.

- The synaptic weights $w_1, w_2, \ldots, w_m$ of the perceptron can **be adapted on an iteration by-iteration basis.**

PERCEPTRON-error correction learning rule

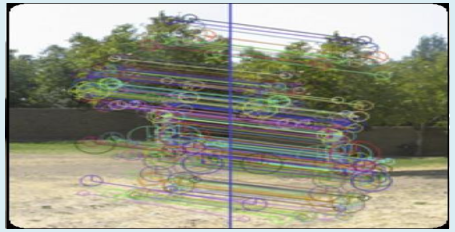
❖ Adaptation of the weight vector **w**(*n*)

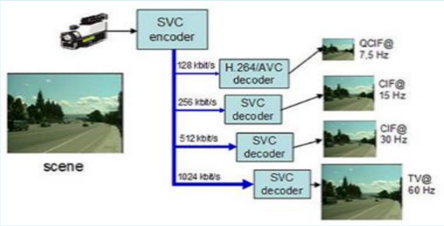
LMS algorithm: $\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}[n] + \mu \mathbf{u}[n] e^*[n].$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta[d(n) - y(n)]\mathbf{x}(n)$$: ***error-correction learning rule***

where $d(n) = \begin{cases} +1 & \text{if } \mathbf{x}(n) \text{ belongs to class } \mathscr{C}_1 \\ -1 & \text{if } \mathbf{x}(n) \text{ belongs to class } \mathscr{C}_2 \end{cases}$, η is the *learning-rate parameter* and the difference *d*(*n*) - *y*(*n*) plays the role of an *error signal.*

It controls the adjustment applied to the weight vector at iteration *n.*





Contents

- Multilayer Perceptron (MLP)
- Back-Propagation Algorithm (BPA)
- **Regression Analysis**

❖ Regression analysis

- A **statistical process for estimating the relationships among variables**.
- Sometimes, *parameter estimation* problem!!!

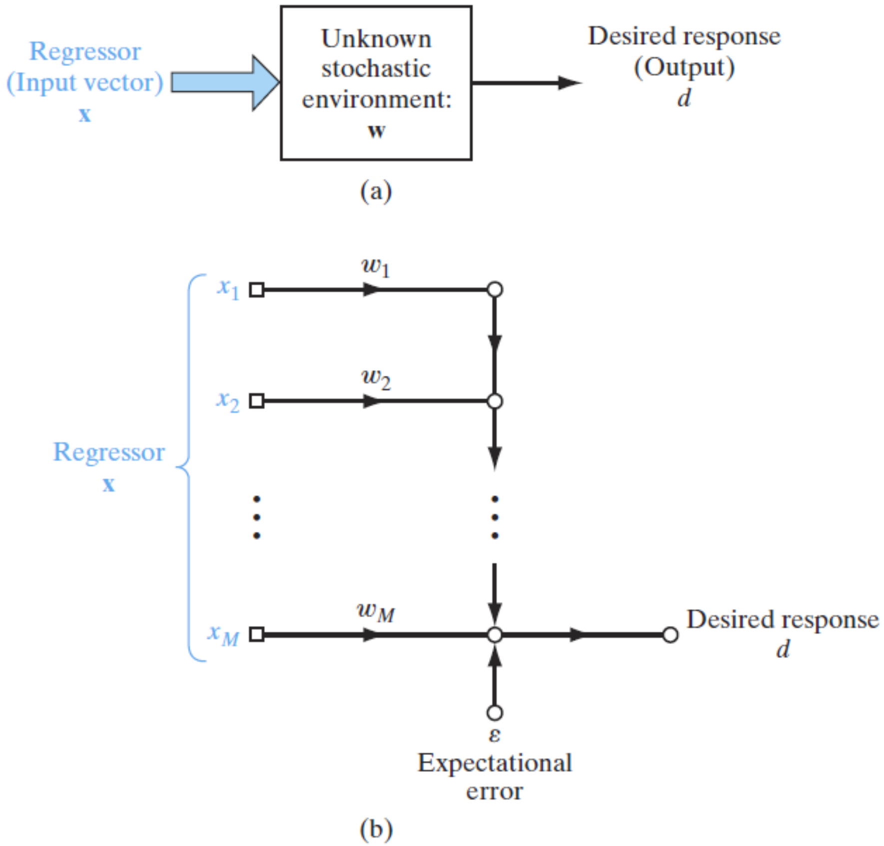
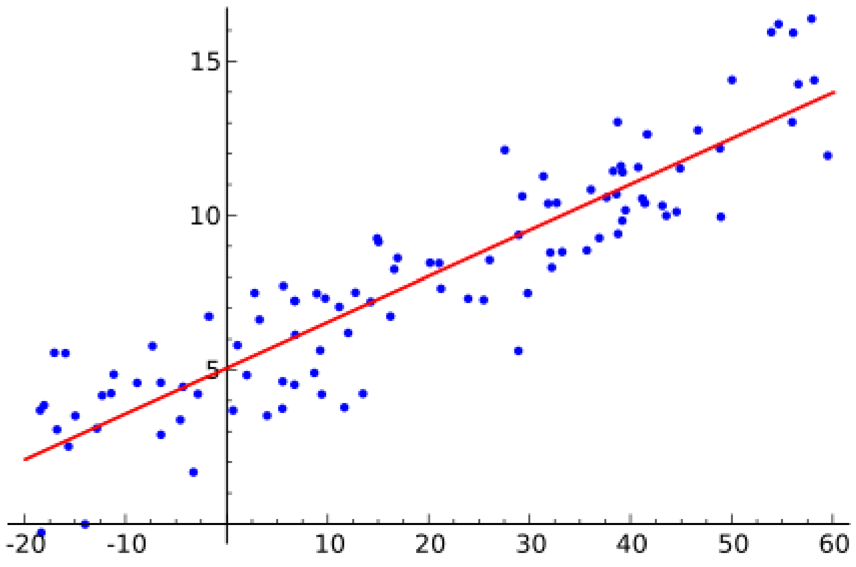
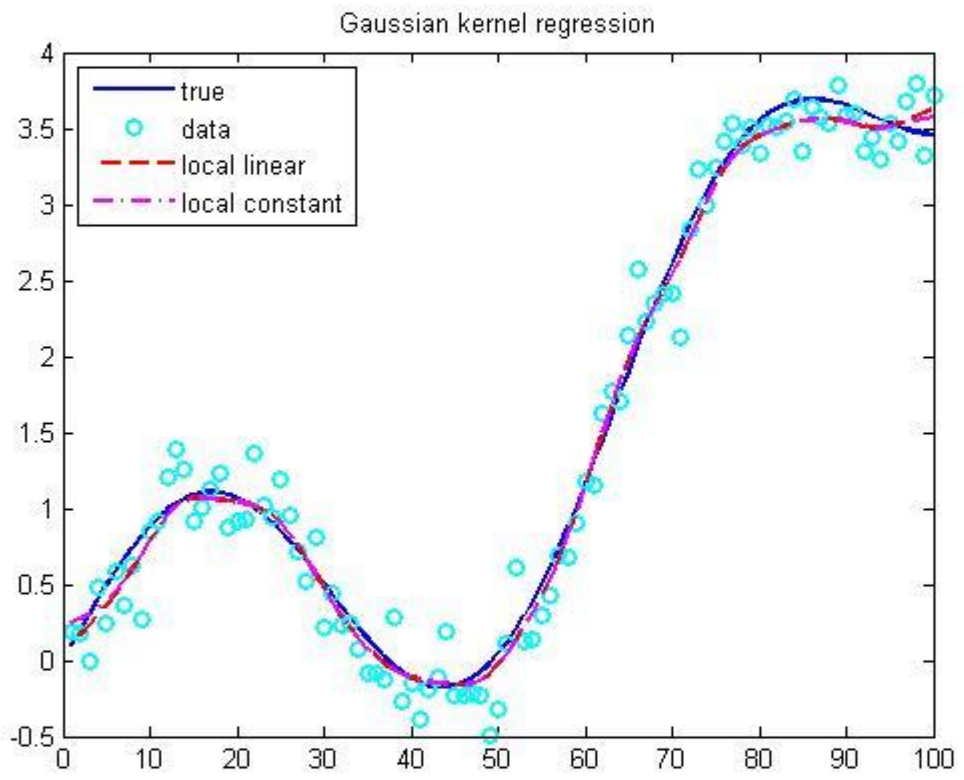


FIGURE 2.1 (a) Unknown stationary stochastic environment. (b) Linear regression model of the environment.

Given the joint statistics of the regressor ***X*** *and the corresponding response D, estimate the unknown parameter vector* **w**.

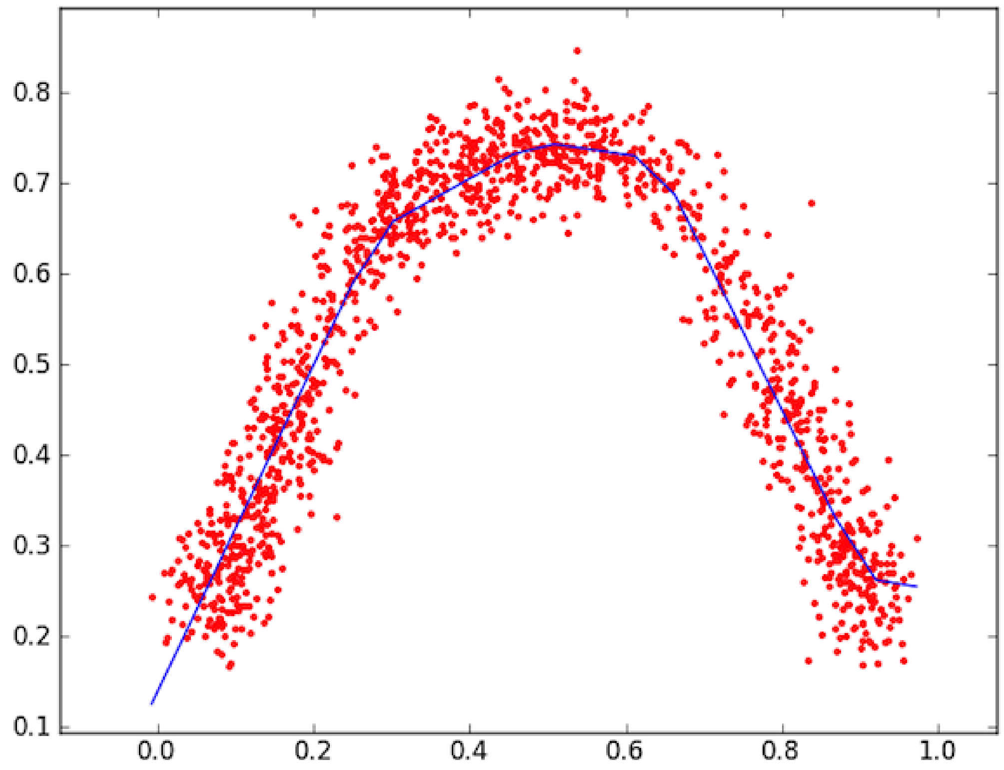
- **Data fitting (Curve fitting)** is a good example of regression analysis.

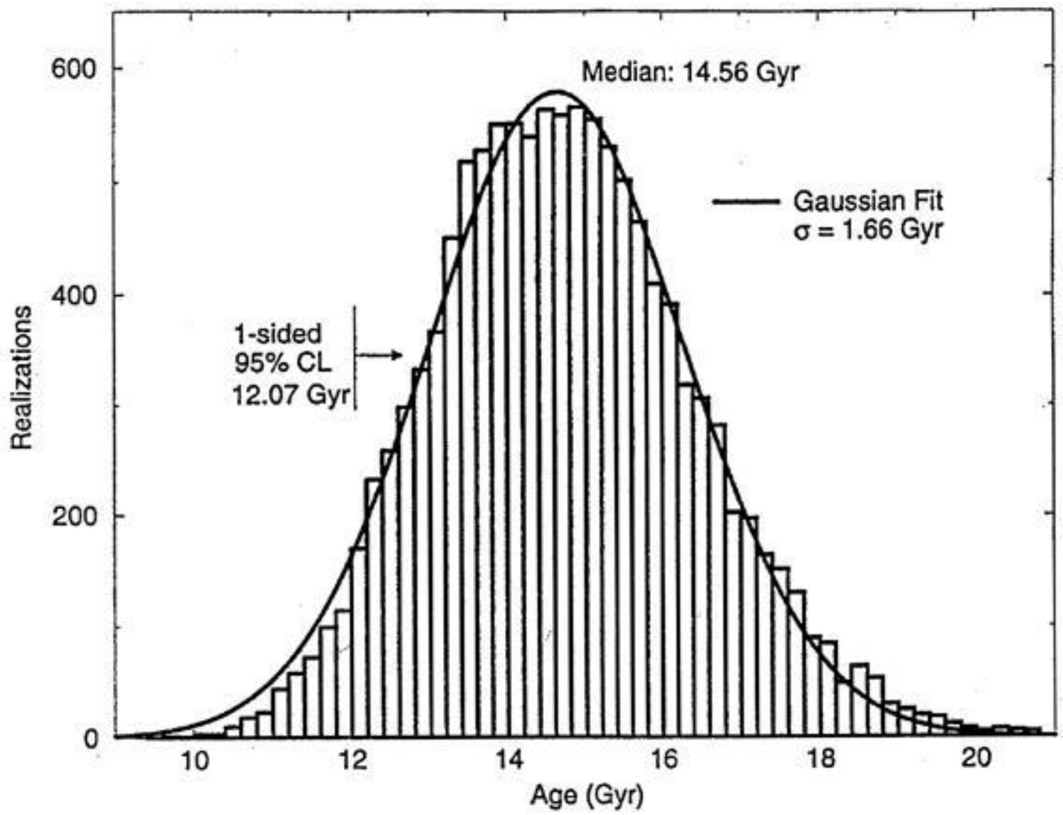




Regression Analysis

- **Data fitting (Curve fitting)** is a case of regression analysis.





Thank you for your attention.!!!
QnA

http://ivpl.sookmyung.ac.kr