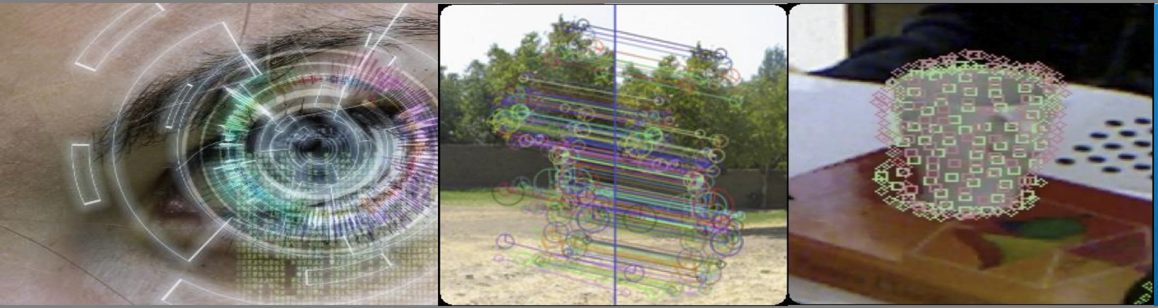
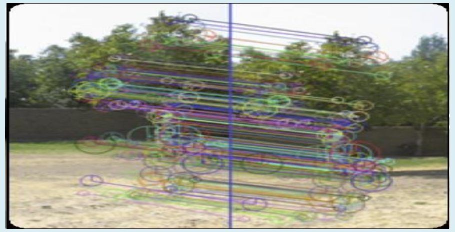
Computer Vision

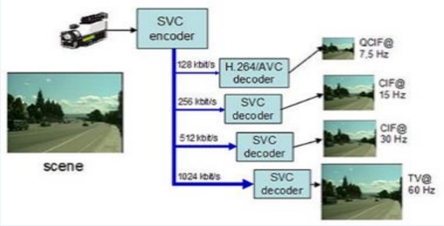
Deep Learning: Artificial Neural Networks (ANN)(1)
- A Brief Introduction: Concept and Structures



2023 Fall

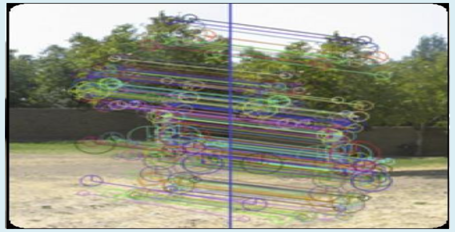
Prof. Byung-Gyu Kim
Intelligent Vision Processing Lab. (IVPL)
https://ivpl.sookmyung.ac.kr
Dept. of IT Engineering, Sookmyung Women's University
E-mail: bg.kim@ivpl.sookmyung.ac.kr

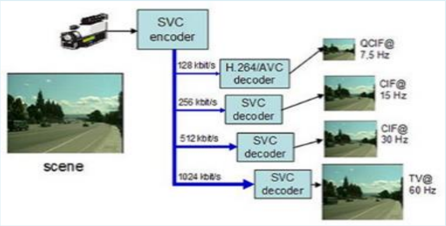




Contents

- GLOSSARY
- What is Neural Networks?
- Steepest Gradient Decent Algorithm
- Regression Analysis





Contents

- **GLOSSARY**
- **What is Neural Networks?**
- **Steepest Gradient Decent Algorithm**
- Regression Analysis

❖ NOTATIONS I: MATRIX ANALYSIS

- **Scalars:** Italic lowercase symbols are used for scalars.
- **Vectors:** Bold lowercase symbols are used for vectors.
- A vector is defined as a *column* of scalars. Thus, the *inner product* of a pair of *m* dimensional vectors, **x** and **y**, is written as

$$\mathbf{x}^T\mathbf{y} = [x_1, x_2, ..., x_m]\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

$$= \sum_{i=1}^{m} x_i y_i$$

where the superscript *T* denotes *matrix transposition*. With the inner product being a scalar, we therefore have

$$\mathbf{y}^T\mathbf{x} = \mathbf{x}^T\mathbf{y}$$

GLOSSARY

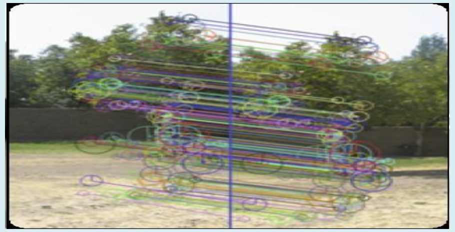
- **Matrices:** Bold uppercase symbols are used for matrices.
- Matrix multiplication is carried out on a *row multiplied by column basis*. To illustrate, consider an *m*-by-*k* matrix **X** and a *k*-by-*l* matrix **Y**. The product of these two matrices yields the *m*-by-*l* matrix

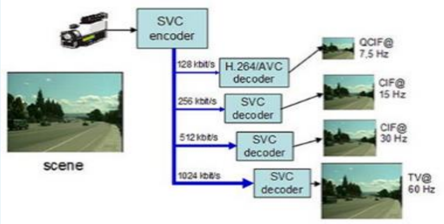
$$\mathbf{Z} = \mathbf{XY}$$

- The outer product of a pair of *m*-dimensional vectors, **x** and **y**, is written as $\mathbf{xy}^T$, which is an *m*-by-*m* matrix.

❖ NOTATIONS II: PROBABILITY THEORY

- **Random variables:** Italic uppercase symbols are used for random variables. The sample value (i.e., one-shot realization) of a random variable is denoted by the corresponding italic lowercase symbol. For example, we write X for a random variable and x for its sample value.
- **Random vectors:** Bold uppercase symbols are used for random vectors. Similarly, the sample value of a random vector is denoted by the corresponding bold lowercase symbol. For example, we write **X** for a random vector and **x** for its sample value.
- The ***probability density function*** **(pdf)** of a random variable **X** is thus denoted by $p_{\mathbf{X}}(\mathbf{x})$, which is a function of the sample value **x**; the subscript **X** is included as a reminder that the pdf pertains to random vector **X**.





Contents

- GLOSSARY
- **What is Neural Networks?**
- Steepest Gradient Decent Algorithm
- Regression Analysis

What is Neural Network?(0)

❖ Human Brain

What is Neural Network?(1)

❖ Cell of Human Brain

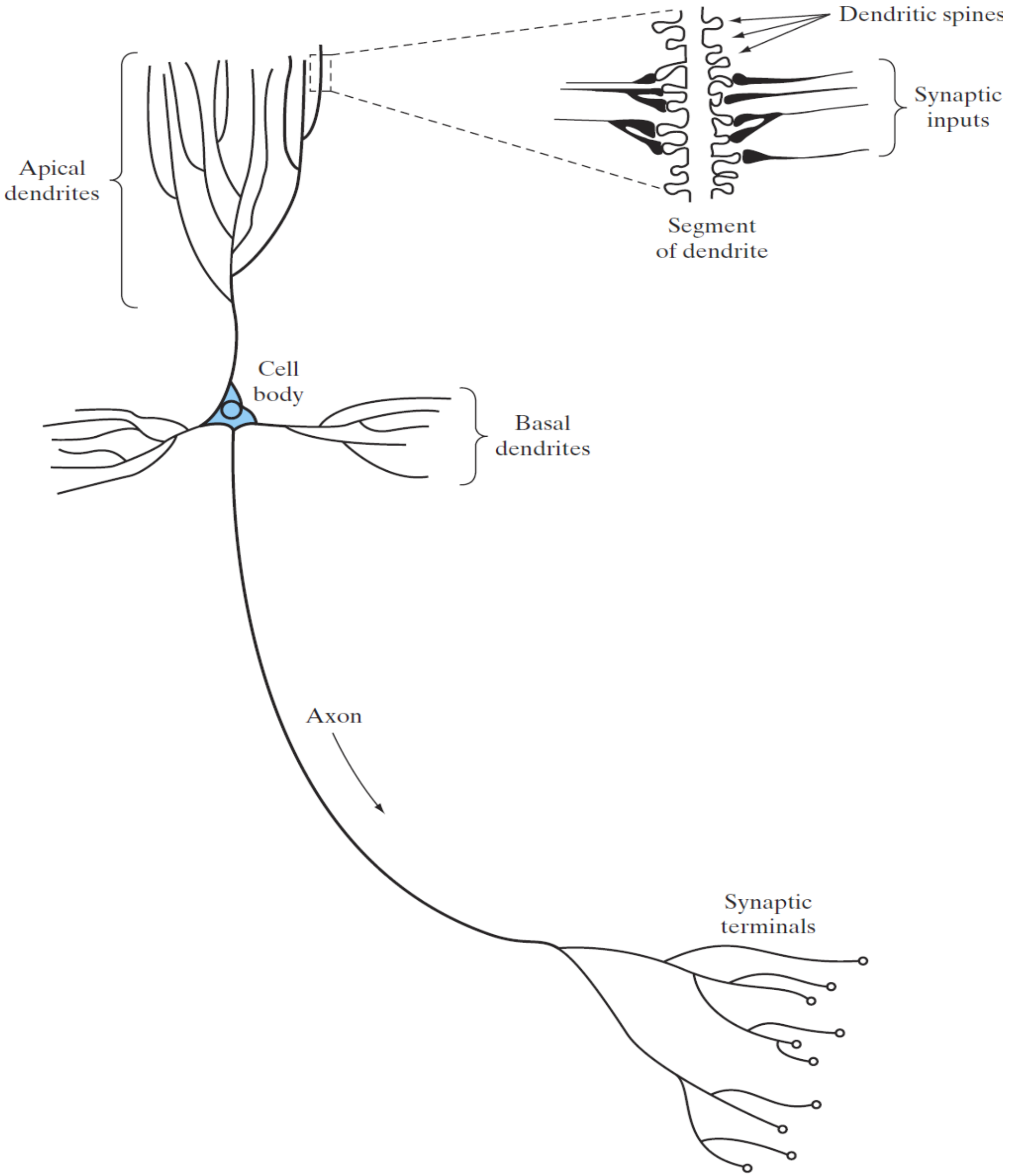


FIGURE 2 The pyramidal cell.

IVPL Intelligent Vision Processing Lab

What is Neural Network?(1)

- The **human brain** computes in an entirely different way from **the conventional digital computer**.
- The brain is a highly complex, nonlinear, and parallel computer (information-processing system).
  - It has the capability to organize **its structural constituents**, known as ***neurons***, so as to perform certain computations (e.g., pattern recognition, perception, and motor control) many times faster than the fastest digital computer in existence today.
  - How, then, does a human brain or the brain of a bat do it?
  - At birth, a brain already has **considerable structure** and **the ability** to build up its own rules of behavior **through what we usually refer to as "*experience*."**

What is Neural Network?(2)

❖ Definition

- A **neural network** is a massively parallel distributed processor made up of simple processing units that has a natural propensity for storing **experiential knowledge** and making it available for use. It resembles the brain in two respects:

1. Knowledge is acquired by the network from its environment through a **learning process.**

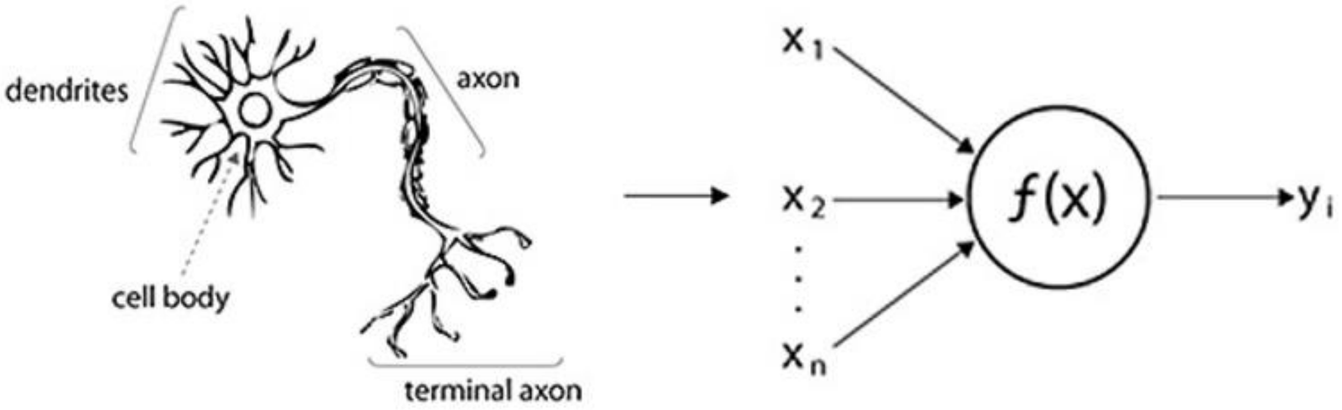
2. **Interneuron connection strengths**, known as ***synaptic weights***, are used **to store the acquired knowledge**.

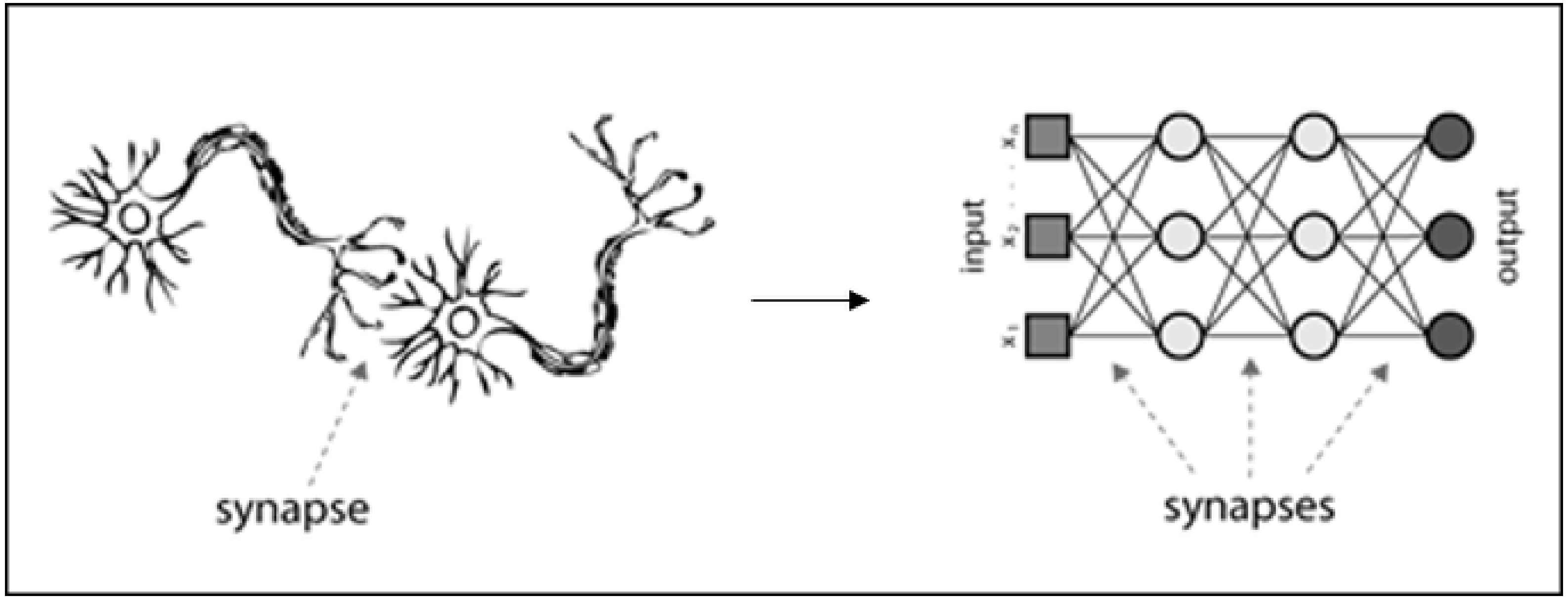
❖ The human nervous system

FIGURE 1 Block diagram representation of nervous system.

- The **receptors**: convert stimuli from the human body or the external environment into electrical impulses that convey information to the neural net (brain).
- The **effectors**: convert electrical impulses generated by the neural net into discernible responses as system outputs.
- The idea of **neurons** as *structural constituents of the brain*.
  - Massive interconnections === Networks
  - Synapses, or nerve endings: *elementary structural and functional units* that *mediate the interactions between neurons*.

❖ Neuron Model





MODELS OF A NEURON

❖ **Neuron**: An ***information-processing unit*** that is fundamental to the operation of a neural network.

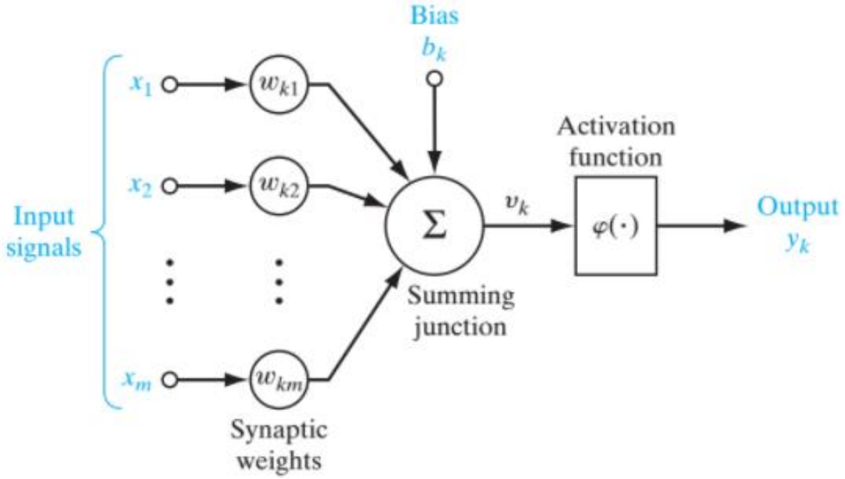


FIGURE 5 Nonlinear model of a neuron, labeled k.

❖ **Three basic elements** of the neural model:

- **A set of synapses, or connecting links**-by a weight or strength of its own.
- **An adder** for summing the input signals.
- **An activation function** for limiting the amplitude of the output of a neuron.

MODELS OF A NEURON

- For the neuron k

$$u_k = \sum_{j=1}^{m} w_{kj} x_j$$

$$y_k = \varphi(u_k + b_k)$$

Activation function

❖ Types of Activation Function

- **Threshold Function**

$$\varphi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases}$$

where $v_k = \sum_{j=1}^{m} w_{kj} x_j + b_k$ for neuron *k*.

$$y_k = \begin{cases} 1 & \text{if } v_k \geq 0 \\ 0 & \text{if } v_k < 0 \end{cases}$$

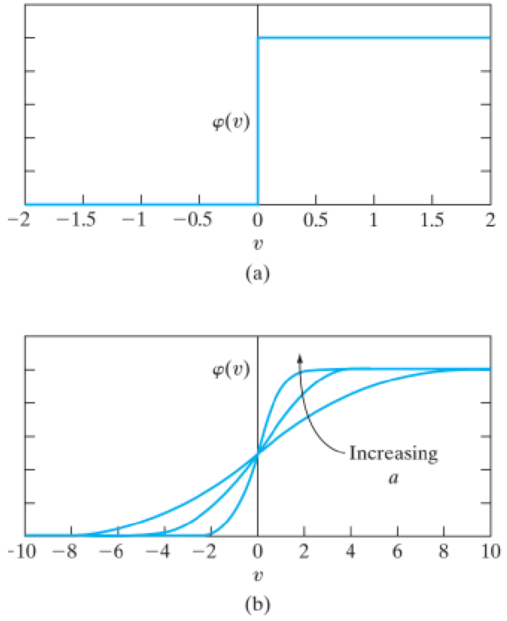


FIGURE 8 (a) Threshold function. (b) Sigmoid function for varying slope parameter *a*.

- **Sigmoid Function**
  - An example of the sigmoid function is the logistic function, defined by

$$\varphi(v) = \frac{1}{1 + \exp(-av)}$$

where a is the slope parameter of the sigmoid function.

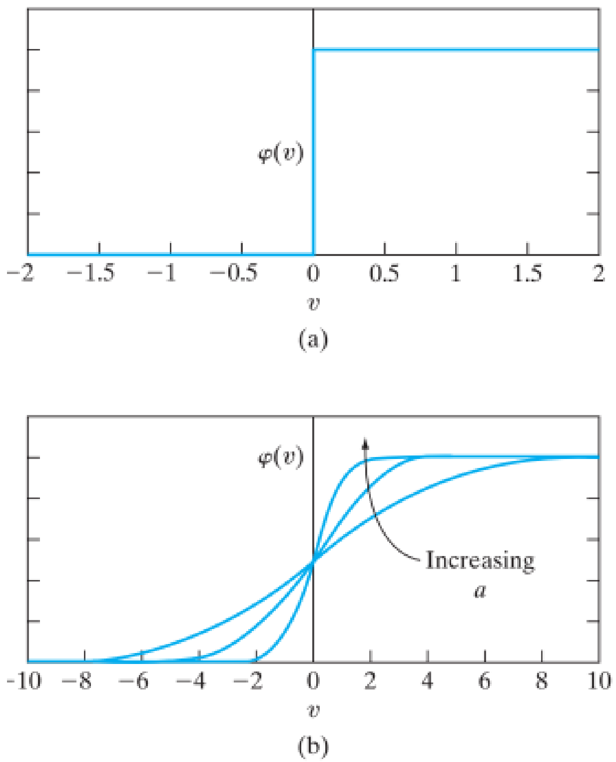
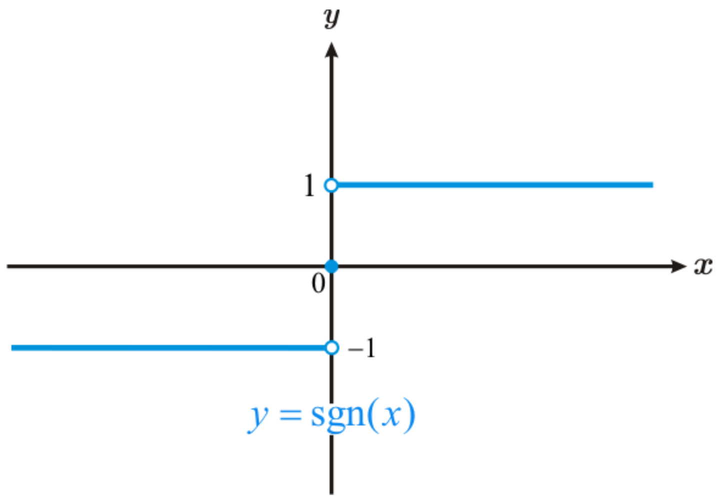


FIGURE 8 (a) Threshold function. (b) Sigmoid function for varying slope parameter a.

- **Signum Function**
  - An example of the signum function is a kind of threshold function, defined by

$$\varphi(v) = \begin{cases} 1 & \text{if } v > 0 \\ 0 & \text{if } v = 0 \\ -1 & \text{if } v < 0 \end{cases}$$



- hyperbolic tangent function

$$\varphi(v) = \tanh(v)$$

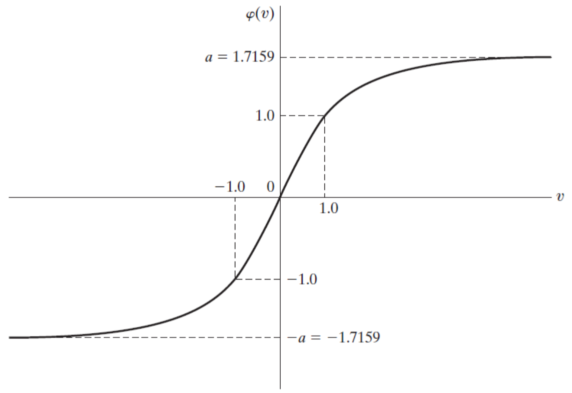
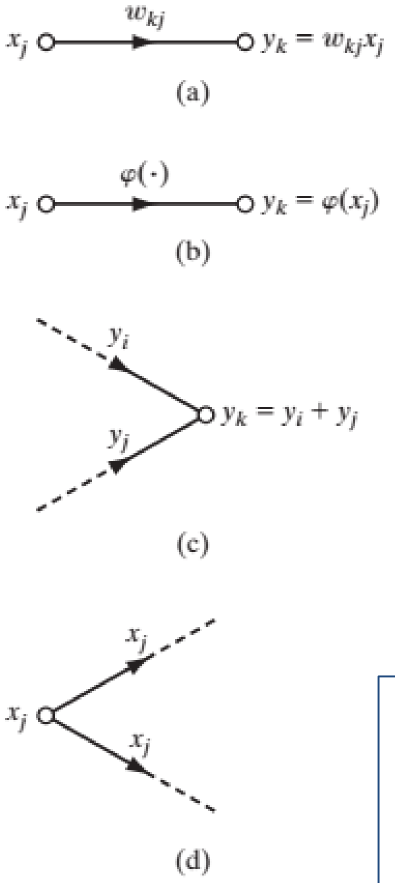


FIGURE 4.10 Graph of the hyperbolic tangent function $\varphi(v) = a\tanh(bv)$ for $a = 1.7159$ and $b = 2/3$. The recommended target values are +1 and −1.

❖ NEURAL NETWORKS VIEWED AS **DIRECTED GRAPHS**

FIGURE 9 Illustrating basic rules for the construction of signal-flow graphs.



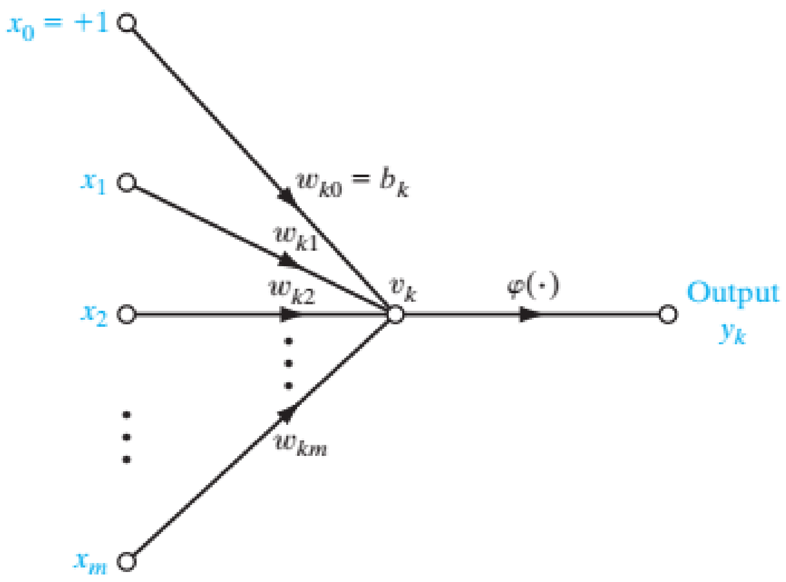


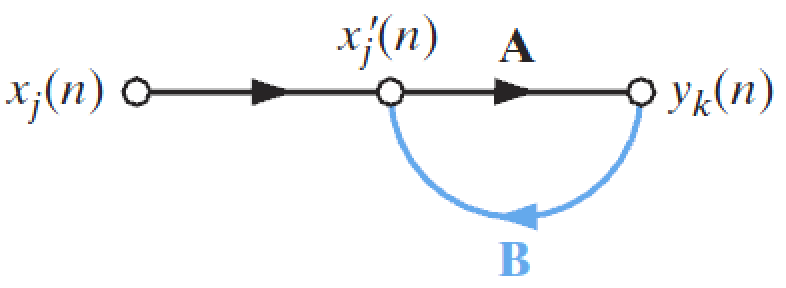
FIGURE 10 Signal-flow graph of a neuron.

1. Source nodes supply input signals to the graph.
2. Each neuron is represented by a single node called a computation node.
3. The communication links interconnecting the source and computation nodes of the graph carry no weight; they merely provide directions of signal flow in the graph.

❖ *Feedback* → ***Recurrent Networks***

- To exist in a dynamic system whenever the output of an element in the system influences in part the input applied to that particular element, thereby giving rise to one or more closed paths for the transmission of signals around the system.

FIGURE 12 Signal-flow graph of a single-loop feedback system.



$$y_k(n) = \mathbf{A}[x_j'(n)]$$

$$x_j'(n) = x_j(n) + \mathbf{B}[y_k(n)]$$

$$y_k(n) = \frac{\mathbf{A}}{1 - \mathbf{AB}}[x_j(n)]$$

NETWORK ARCHITECTURES

❖ In a *layered* neural network, **the neurons are organized in the form of layers**.

❖ Single-Layer Feedforward Networks

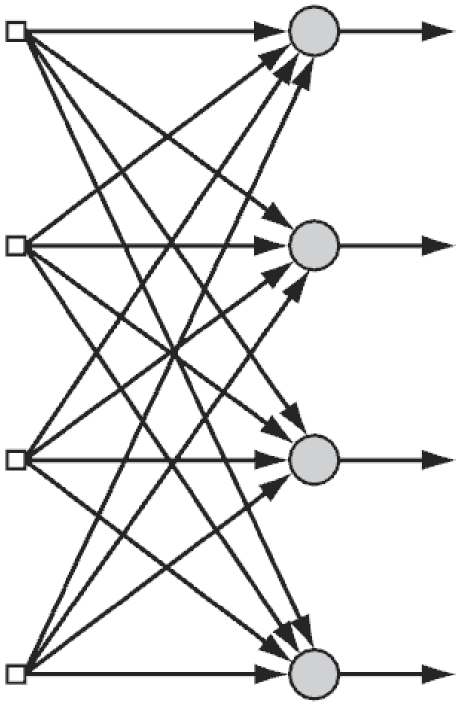


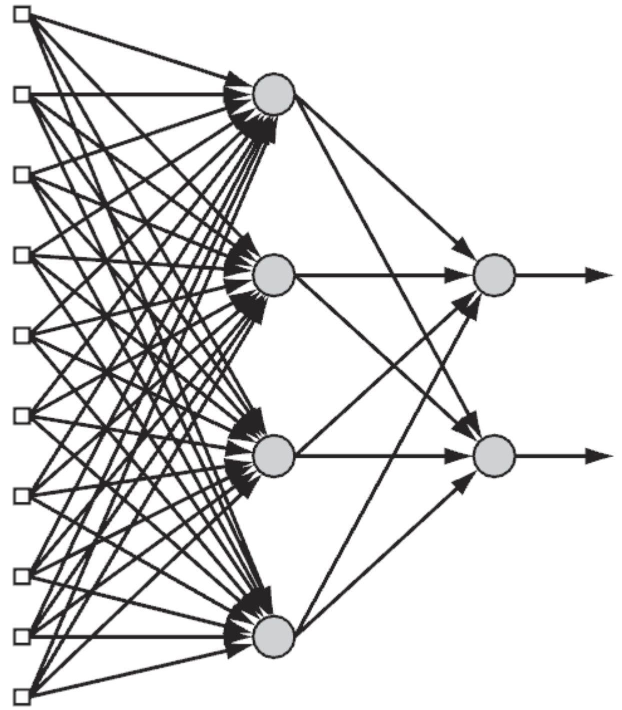
FIGURE 15 Feedforward network with a single layer of neurons.

❖ Multilayer Feedforward Networks

- **Hidden layers (hidden units)**
- This part of the neural network **is not seen directly from either the input or output of the network**.

10 – 4 – 2 network

FIGURE 16 Fully connected feedforward network with one hidden layer and one output layer.



- ***fully connected:***

in the sense that every
node in each layer of the network is
connected to every other node in the adjacent forward layer.

- ***Partially connected:***

some of the communication links (synaptic connections) are
missing from the network,

❖ Recurrent Networks

- A *recurrent neural network* distinguishes itself from a feedforward neural network in that **it has at least one *feedback* loop**.

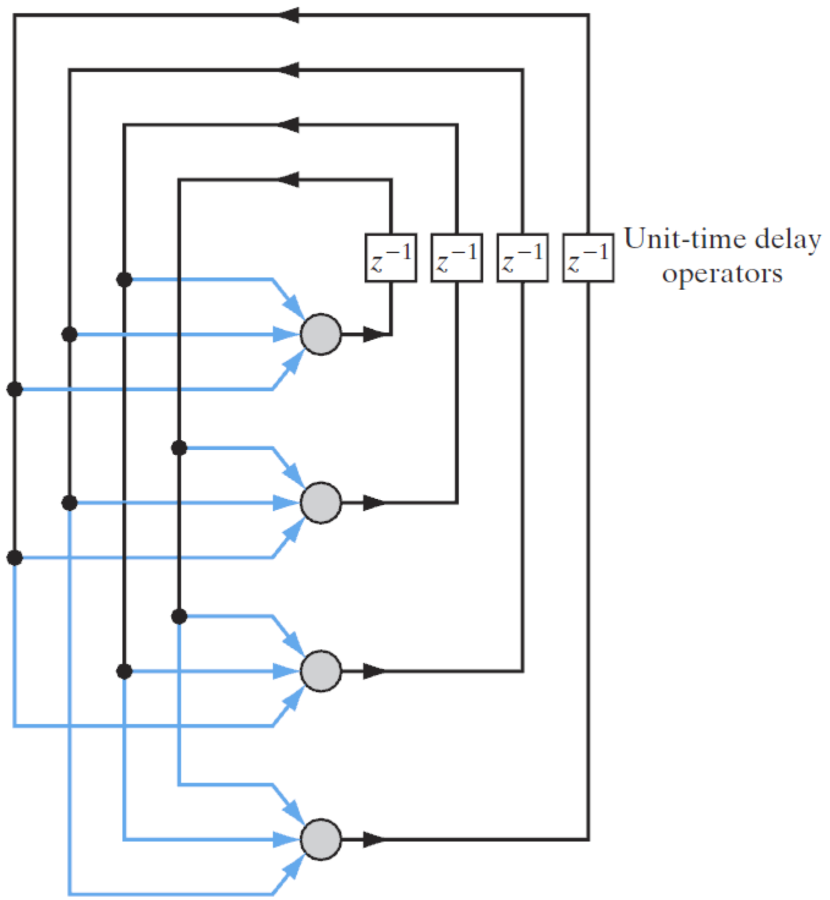


FIGURE 17 Recurrent network with no self-feedback loops and no hidden neurons.

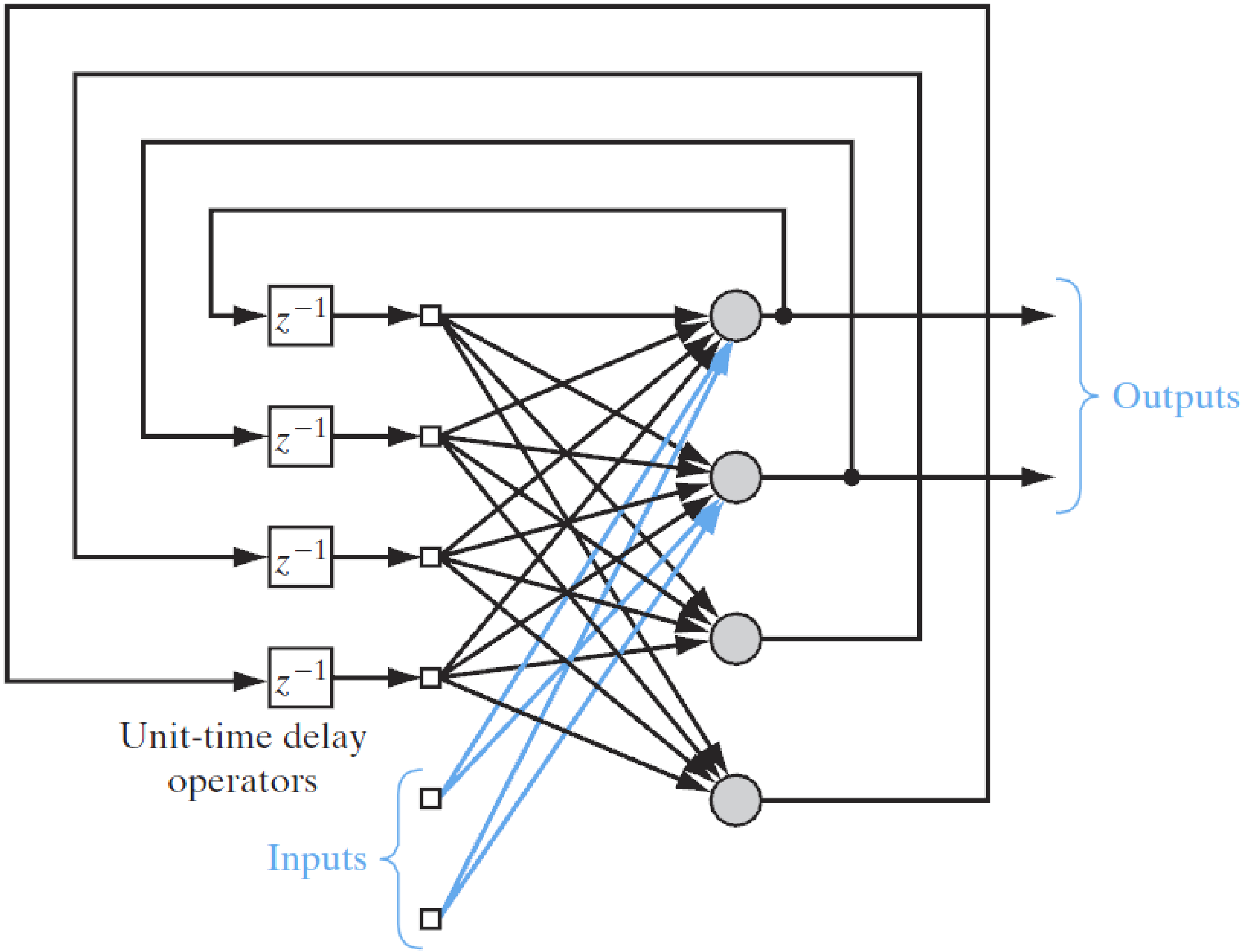


FIGURE 18 Recurrent network with hidden neurons.

KNOWLEDGE REPRESENTATION

- "knowledge" in the definition of a neural network
- A major task for a neural network
  - **is to learn a model of the world (environment) in which it is embedded**, and to maintain the model sufficiently consistently with the real world so as to achieve the specified goals of the application of interest.

Prior information ⟷ Observations (Measurement)

  - ***Labeled*** or ***unlabeled*** examples
    - Labeled example: each example representing an *input signal* is paired with a corresponding *desired response* (i.e., target output). → Set of training data (training sample)
    - Unlabeled example: otherwise.

❖ Design of a neural network

- 1) ***learning*** : A subset of examples is then used to train the network by means of a suitable algorithm
- 2) ***testing (Generalization)*** : will be tested **with data not seen before**.

- How knowledge is actually represented inside an artificial network?
  - there are **four rules for knowledge representation** that are of a general commonsense nature,

Rule 1. Similar inputs (i.e., patterns drawn) from **similar** classes should usually produce similar representations inside the network, and should therefore be classified as belonging to the same class.

  - *measure of similarity : **Euclidian distance***

let $\mathbf{x}_i$ denote an m-by-1 vector,

$$\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{im}]^T$$

Euclidian distance: $$d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\| = \left[\sum_{k=1}^{m} (x_{ik} - x_{jk})^2\right]^{1/2}$$

where x_{ik} and x_{jk} are the kth elements of the input vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively.

- ***dot product***, or ***inner product***

Given a pair of vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ of the same dimension, their **inner product** is $\mathbf{x}_i^T \mathbf{x}_j$, defined as the ***projection*** of the vector $\mathbf{x}_i$ onto the vector $\mathbf{x}_j$,

$$(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j = \sum_{k=1}^{m} x_{ik} x_{jk}$$

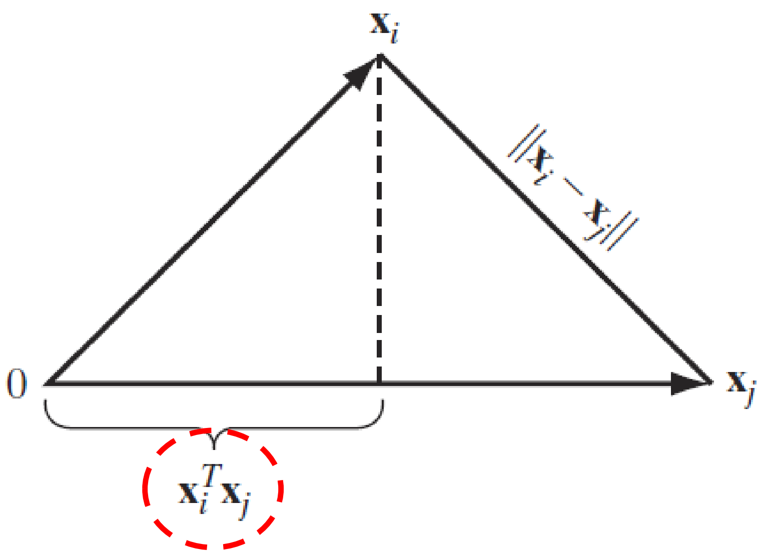


FIGURE 19 Illustrating the relationship between inner product and Euclidean distance as measures of similiarity between patterns.

$$(\mathbf{x}_i, \mathbf{x}_j) = |\mathbf{x}_i||\mathbf{x}_j| \cos\theta$$

- ***Mahalanobis distance***
  - If the vectors **x***i* and **x***j* are ***stochastic***, let μ_i and μ_j denote the mean values of the vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively. That is,

$$\boldsymbol{\mu}_i = \mathbb{E}[\mathbf{x}_i]$$

where $\mathbb{E}$ is the ***statistical expectation operator*** over the *ensemble* of data vectors **x***i*.

- The squared value of this distance from $\mathbf{x}_i$ to $\mathbf{x}_j$ is defined by

$$d_{ij}^2 = (\mathbf{x}_i - \boldsymbol{\mu}_i)^T \mathbf{C}^{-1} (\mathbf{x}_j - \boldsymbol{\mu}_j)$$

where $\mathbf{C}^{-1}$ is the *inverse* of the covariance matrix **C.**

$$\begin{aligned} \mathbf{C} &= \mathbb{E}[(\mathbf{x}_i - \boldsymbol{\mu}_i)(\mathbf{x}_i - \boldsymbol{\mu}_i)^T] \\ &= \mathbb{E}[(\mathbf{x}_j - \boldsymbol{\mu}_j)(\mathbf{x}_j - \boldsymbol{\mu}_j)^T] \end{aligned}$$

- **Correlation/Correlated**
  - the issue of how these two vectors are *correlated* to each other.
  - How much of the overlapped to each other?

- **Uncorrelated/Uncorrelation**
  - There is no overlapped between signals (vectors).
  - The inner product is zero value.

$$(\mathbf{x}_i, \mathbf{x}_j) = |\mathbf{x}_i||\mathbf{x}_j| \cos\theta$$

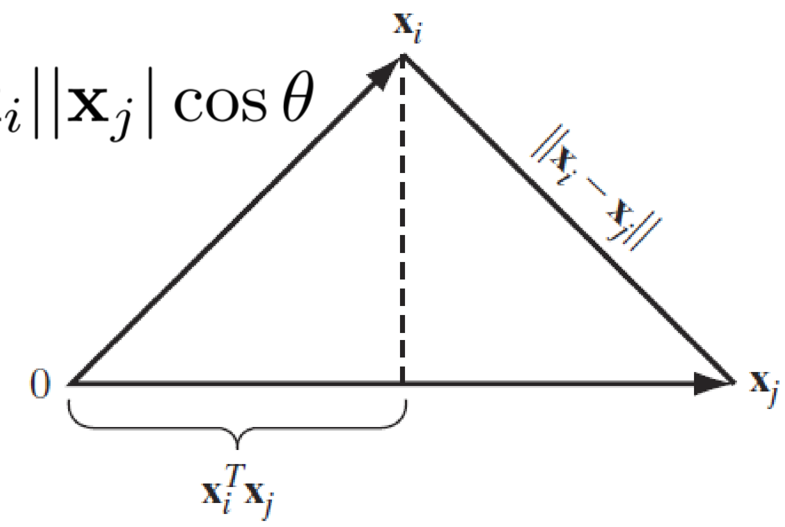


FIGURE 19 Illustrating the relationship between inner product and Euclidean distance as measures of similiarity between patterns.

Rule 2. Items to be categorized as separate classes should be given widely different representations in the network.

Rule 3. If a particular feature is important, then there should be a large number of neurons involved in the representation of that item in the network.

Rule 4. Prior information and invariances should be built into the design of a neural network whenever they are available, so as to simplify the network design by its not having to learn them.

❖ How to Build Prior Information into Neural Network Design

1. ***restricting the network architecture***, which is achieved through **the use of local connections** known as ***receptive fields***,
2. ***constraining the choice of synaptic weights***, which is implemented through **the use of *weight-sharing***.

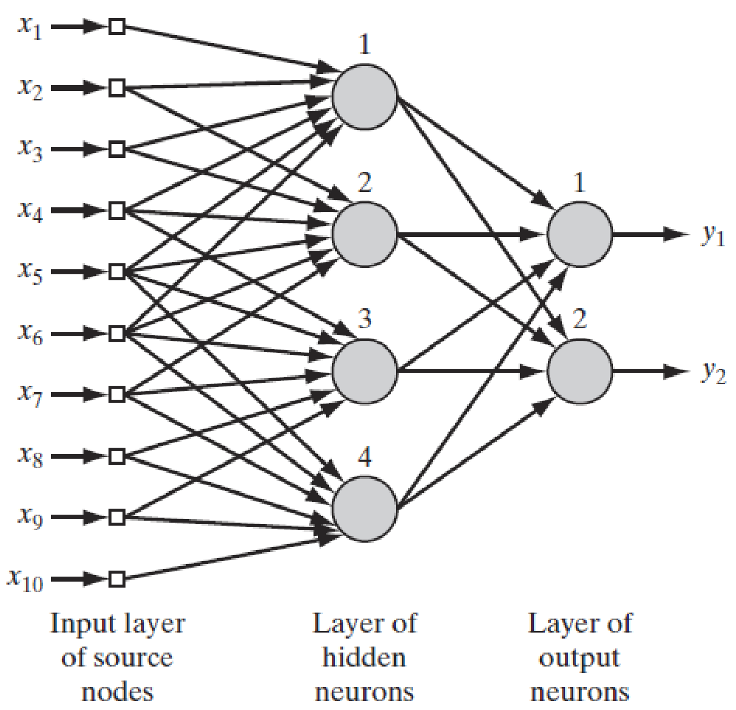


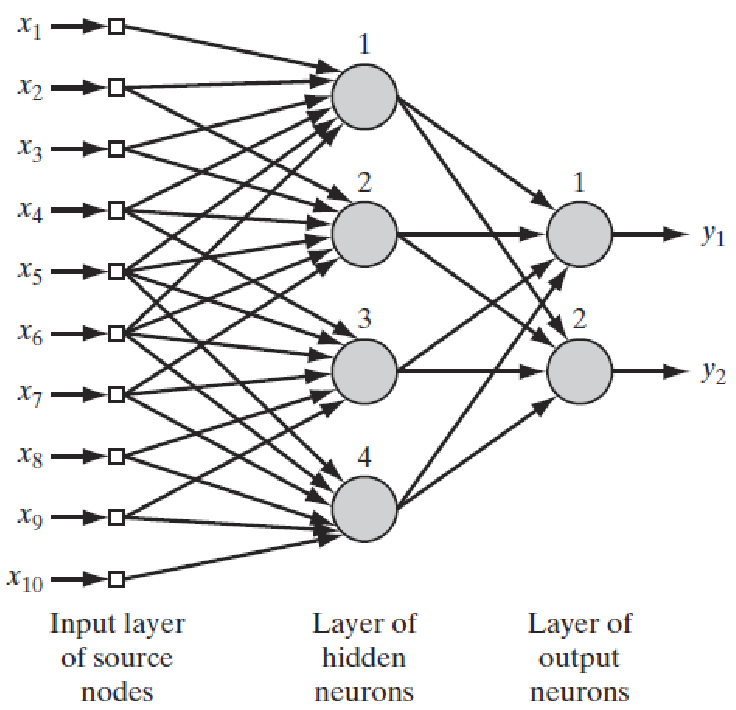
FIGURE 20 Illustrating the combined use of a receptive field and weight sharing. All four hidden neurons share the same set of weights exactly for their six synaptic connections.

[the partially connected feedforward network]

The *receptive* field of a neuron is defined as that region of the input field over which the incoming stimuli can influence the output signal produced by the neuron.

❖ How to Build Prior Information into Neural Network Design

1. ***restricting the network architecture***, which is achieved through **the use of local connections** known as ***receptive fields***;
2. ***constraining the choice of synaptic weights***, which is implemented through **the use of *weight-sharing***.



[the partially connected feedforward network]

FIGURE 20 Illustrating the combined use of a receptive field and weight sharing. All four hidden neurons share the same set of weights exactly for their six synaptic connections.

To satisfy the weight-sharing constraint, the induced local field of hidden neuron *j* as

$$v_j = \sum_{i=1}^{6} w_i x_{i+j-1}, \qquad j = 1, 2, 3, 4$$

where $\{w_i\}_{i=1}^{6}$ constitutes the same set of weights shared by all four hidden neurons, and x_k is the signal picked up from source node $k = i + j - 1$.

→ ***convolution sum***

- Convolutional (Neural) Networks (CNNs)
  - **A feedforward network using *local connections* and *weight sharing*** in the manner described herein is referred to as a ***convolutional network*** (LeCun and Bengio, 2003).

- Learning Processes
  - learning with a teacher (*supervised learning)*
  - learning without a teacher
    - **unsupervised learning**
    - **reinforcement learning**

Learning with a teacher

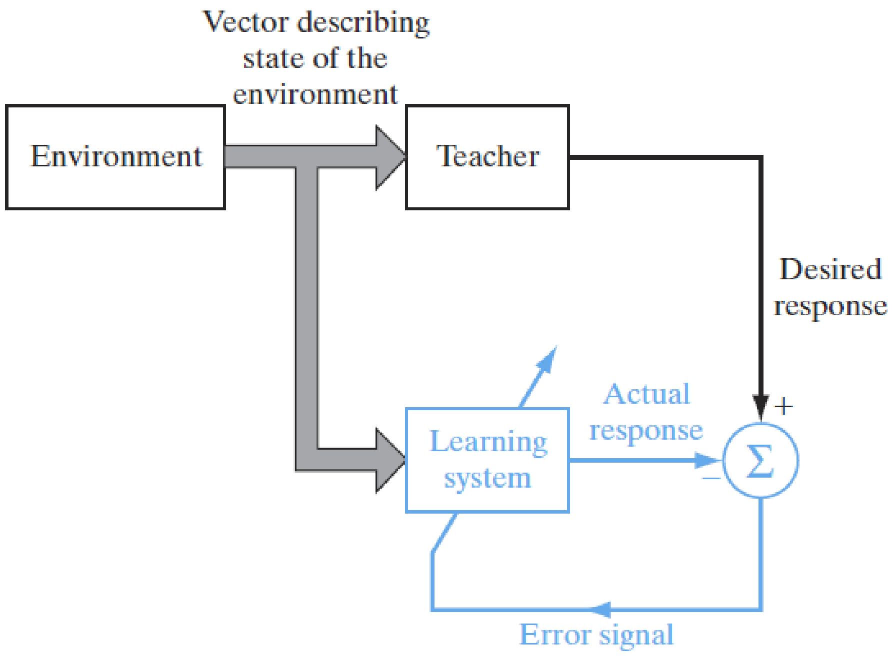


FIGURE 24 Block diagram of learning with a teacher; the part of the figure printed in red constitutes a feedback loop.

- *Error correction learning*
- *error-performance surface*
- *Gradient* of the error surface

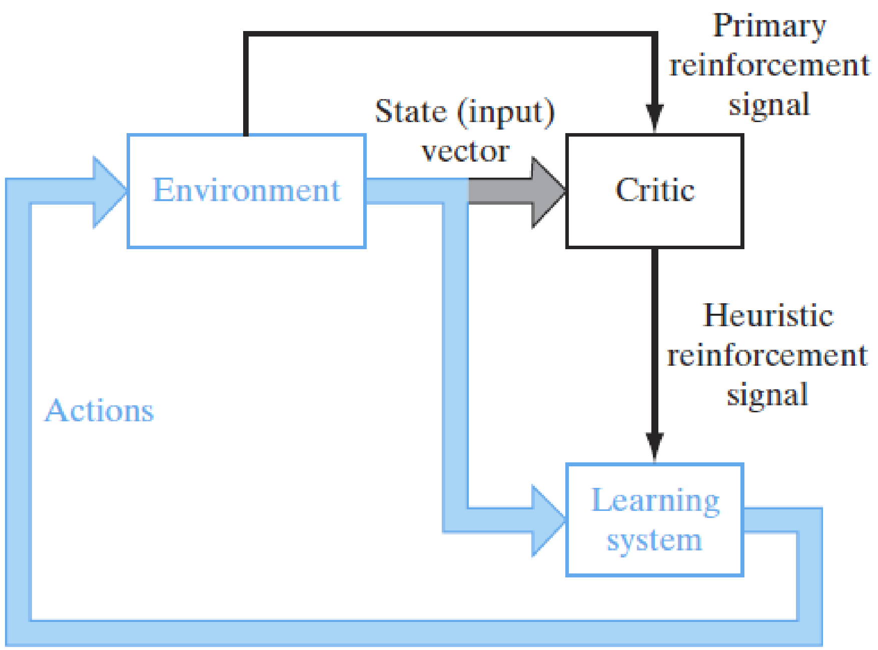
learning without a teacher

there is ***no teacher*** to oversee the learning process.
That is to say, there are **no labeled examples of the function** to be learned by the network.

❖ Reinforcement Learning

The learning of an input–output mapping is performed **through continued interaction with the environment** in order to minimize a scalar index of performance.

FIGURE 25 Block diagram of reinforcement learning; the learning system and the environment are both inside the feedback loop.



❖ **Unsupervised Learning (Self-organized learning)**

- there is ***no external teacher or critic*** to oversee the learning process.

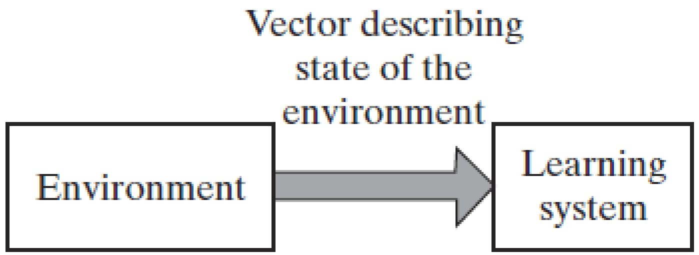
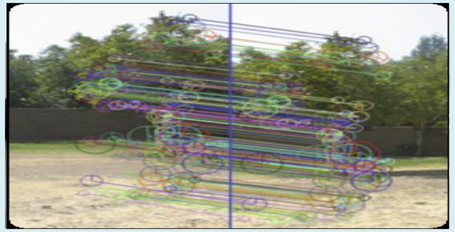
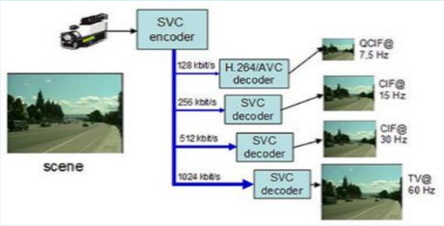


FIGURE 26 Block diagram of unsupervised learning.

- For example, K-means clustering/KNN algorithms and data **clustering with employing the given data's properties**.



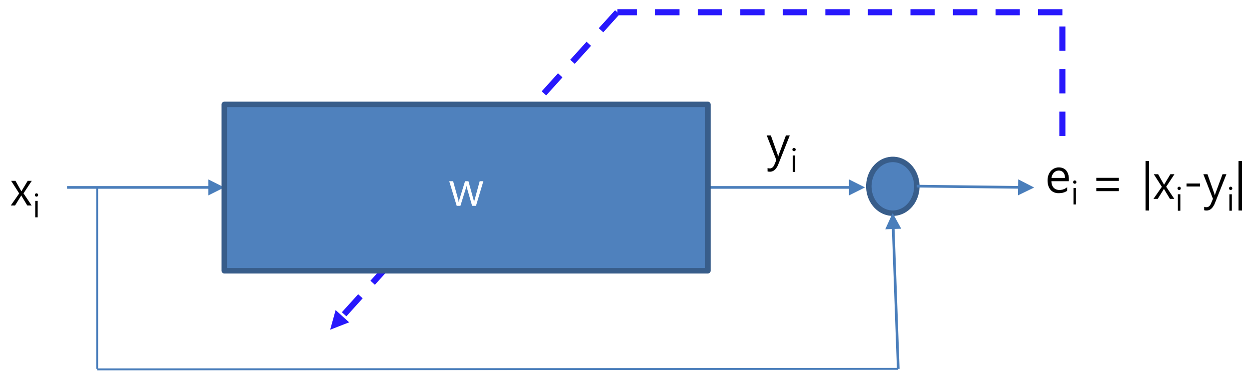


Contents

- GLOSSARY
- What is Neural Networks?
- Steepest Gradient Decent Algorithm
- Regression Analysis

Method of Steepest Descent (1)

- What do we learn???
  - Loss function (L) or Cost function in optimization problem
  - **Problem definition**:
    - We want to make the output y_i as the same one of input x_i .



- *Then how to use the error value to make what we want?*
- *Solution: to minimize the error function or to make zero of the error function*

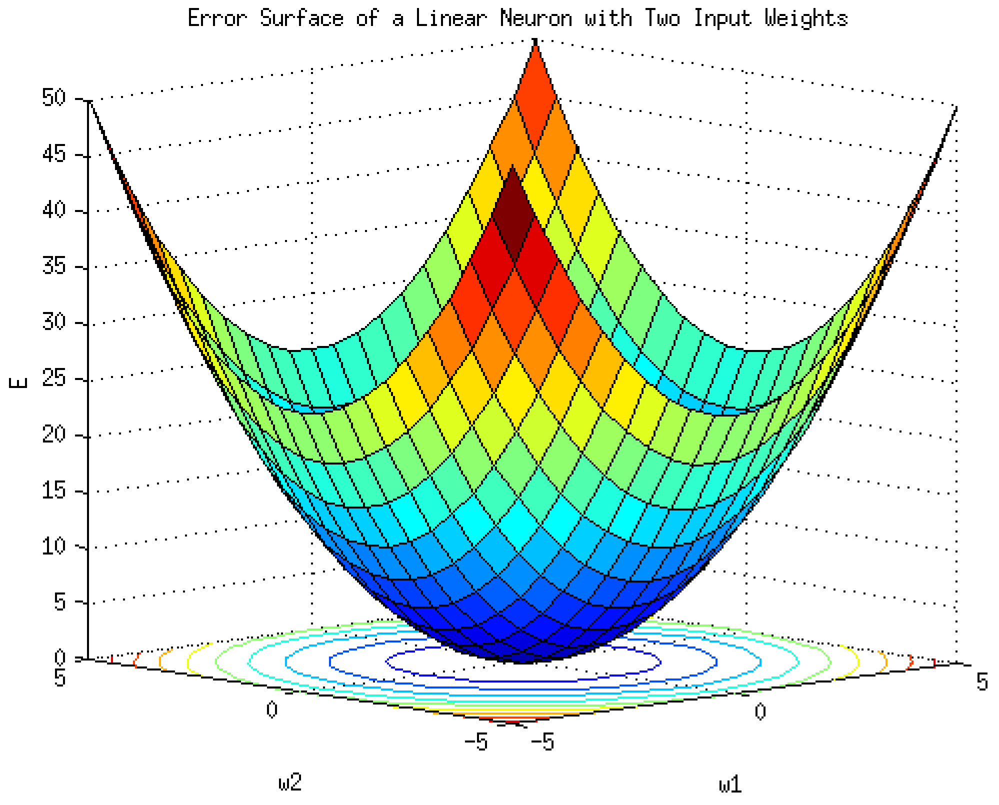
$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{1}{2}\mu[-\nabla J(n)]. \qquad (5)$$

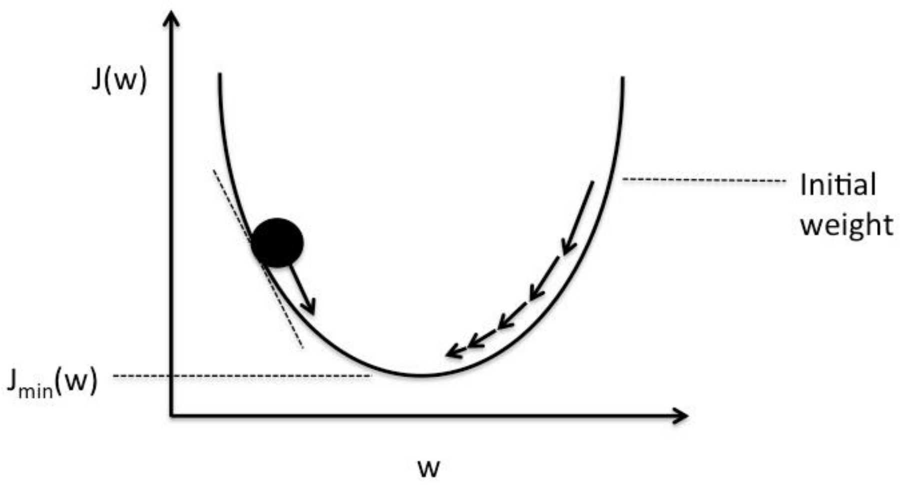
IVPL Intelligent Vision Processing Lab

Method of Steepest Descent (2)

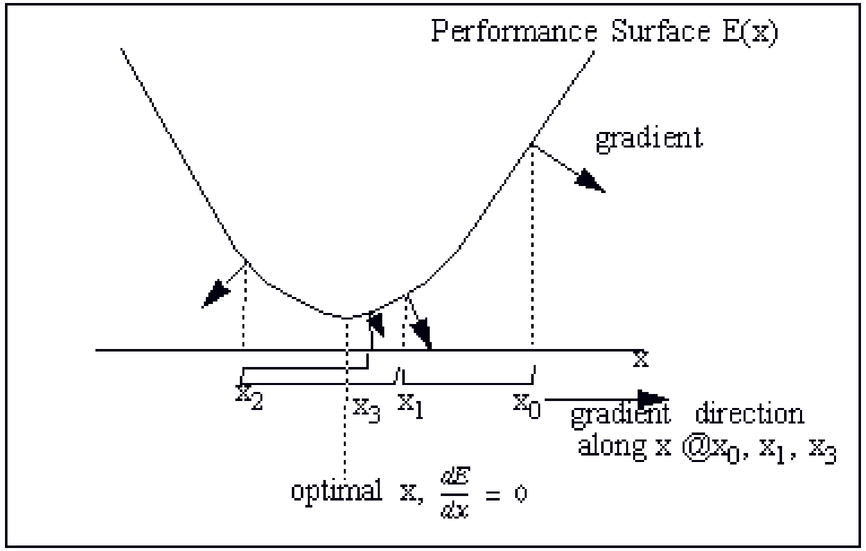
Let's see an example: "**gradient vector**"

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





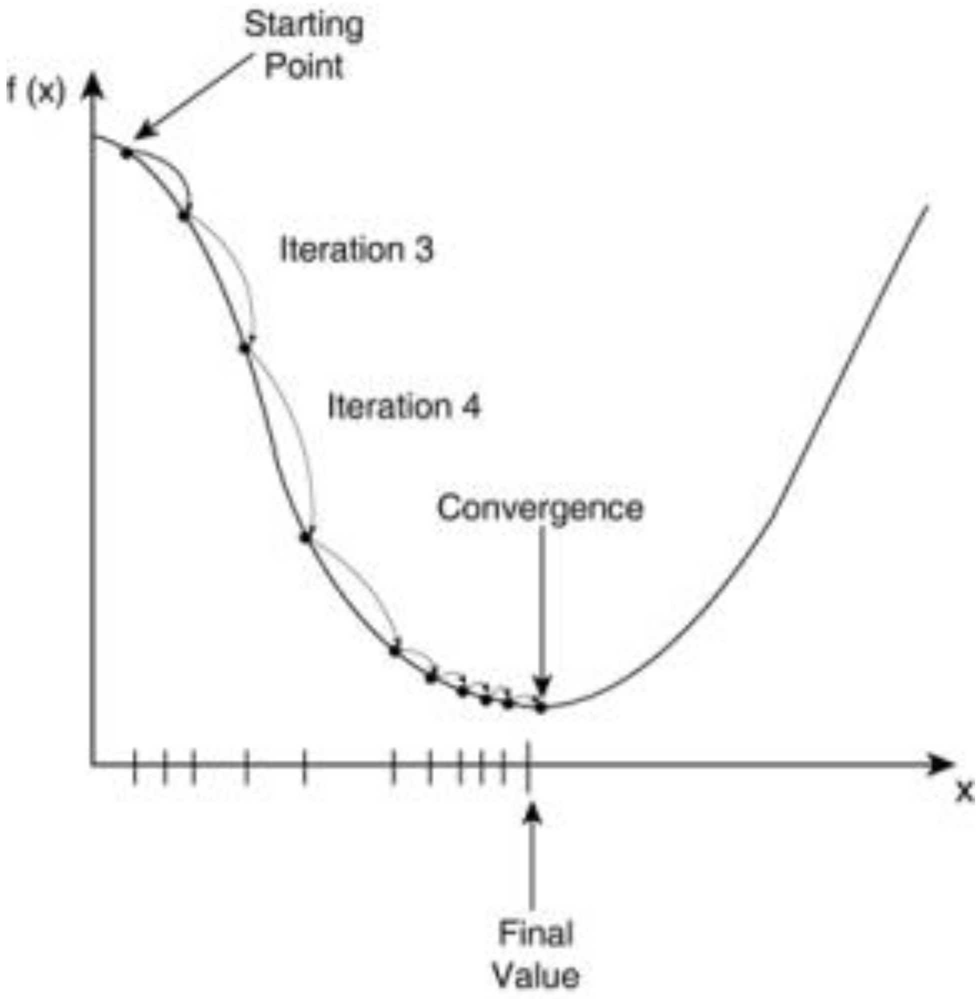
Schematic of gradient descent.



Method of Steepest Descent (3)

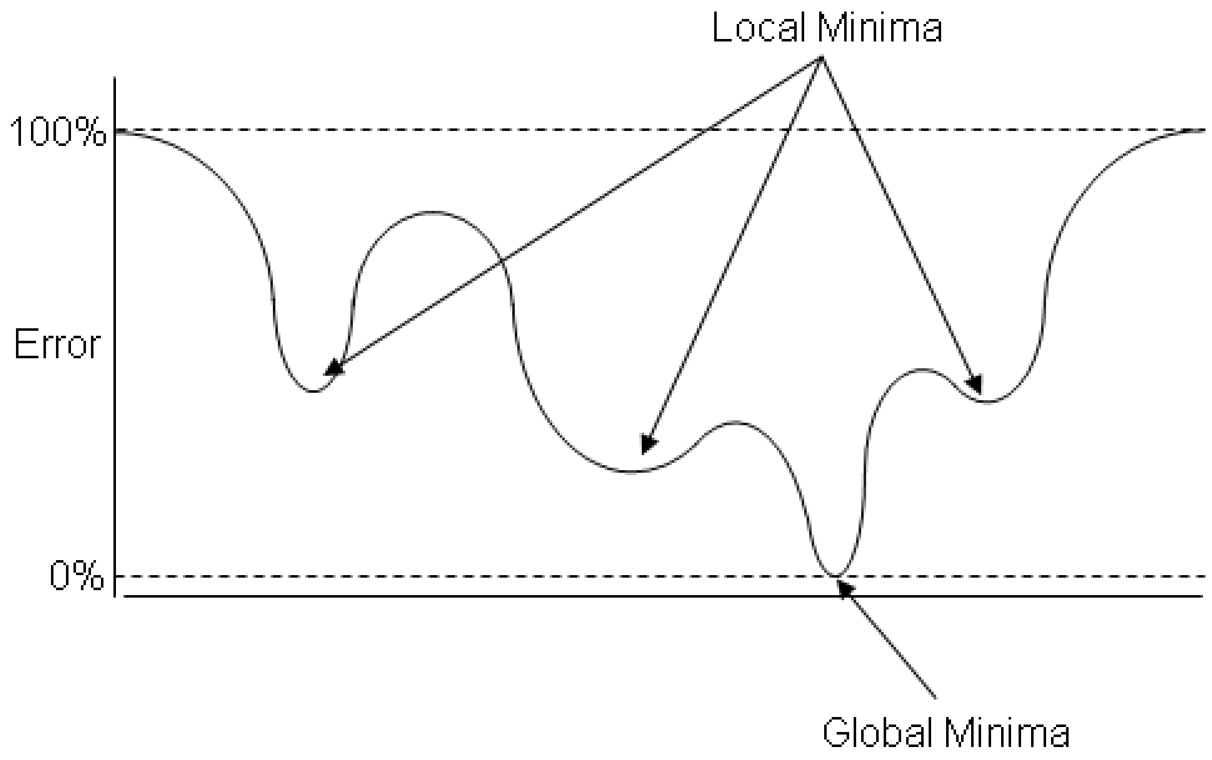
❖ Error Surface: Monotonically Decreasing or Increasing

- One global minimum



Method of Steepest Descent (4)

❖ Error Surface: Non-monotonically Decreasing or Increasing
- Many local minima (minimums)
- One global minimum



❖ Steepest-Descents Path

- *Perpendicular to tangent slope at each point.*

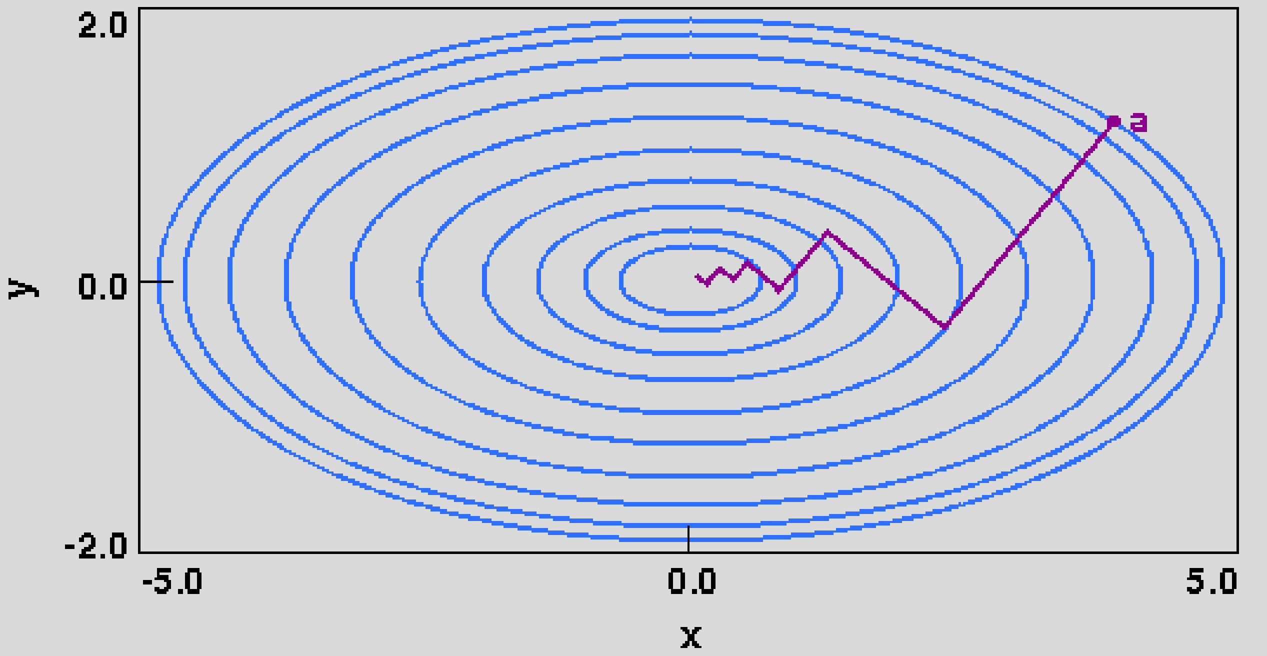
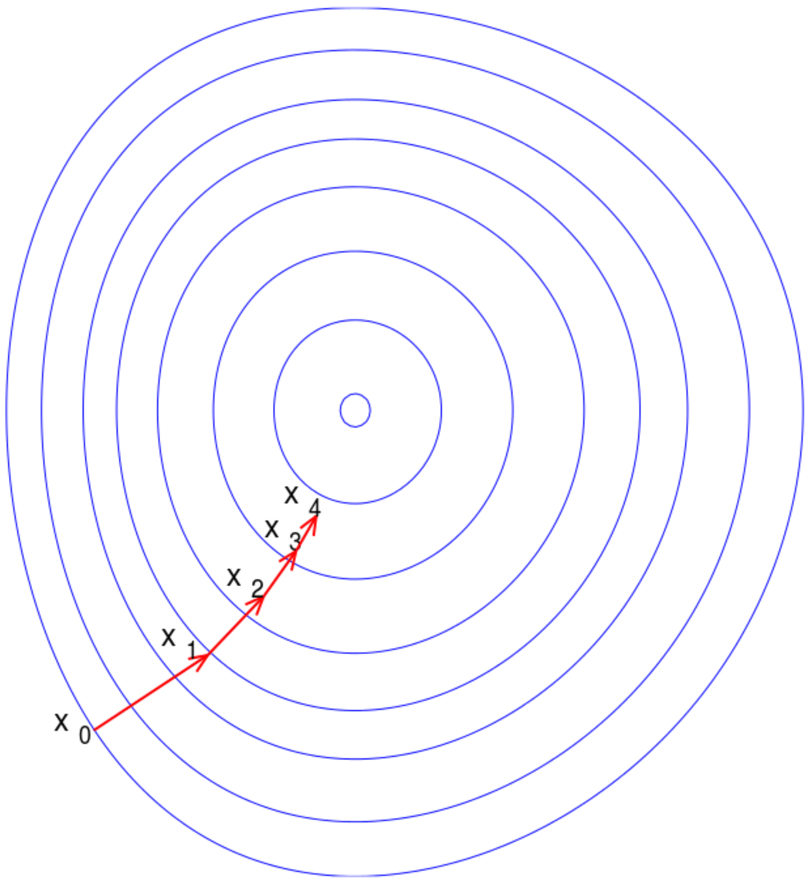
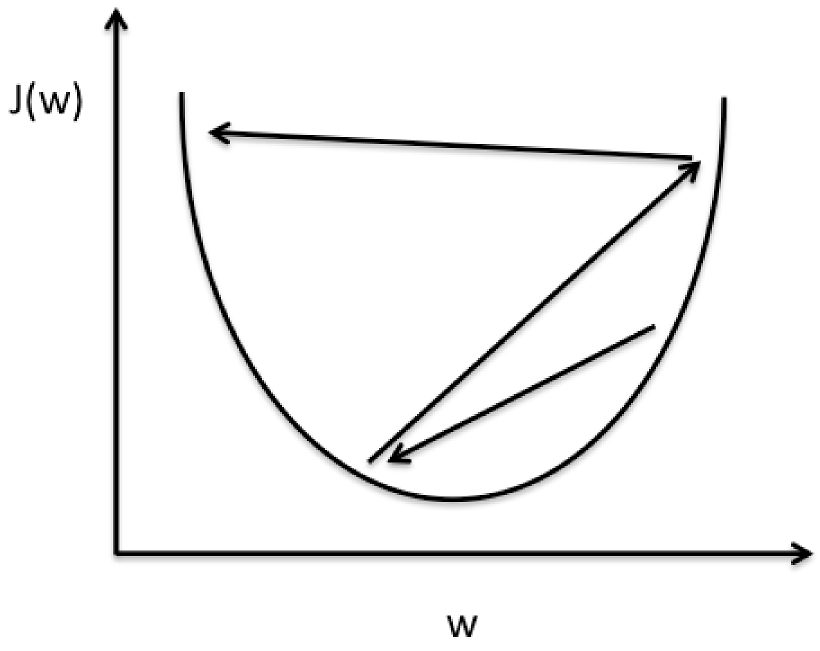


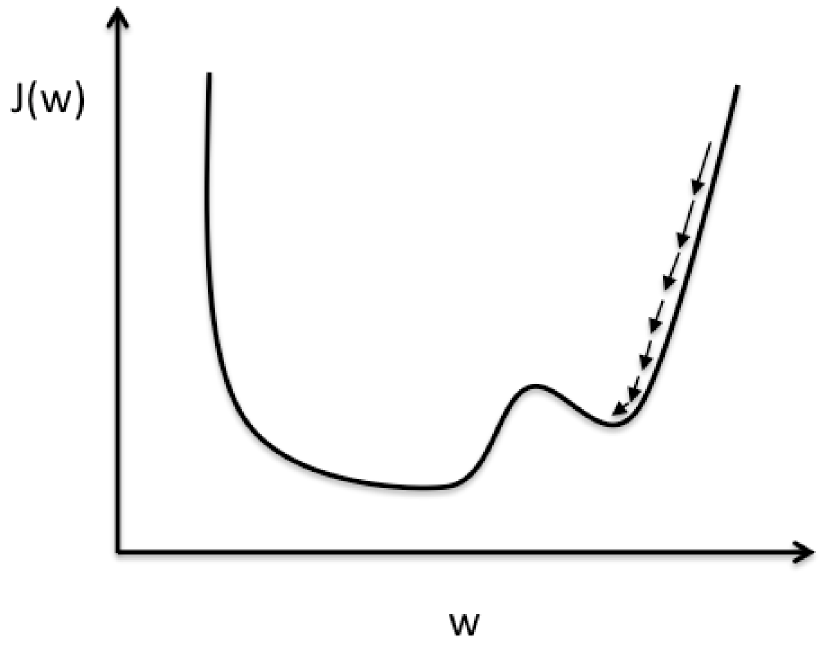
Figure 4–4. Minimization Path following a Steepest-Descents Path



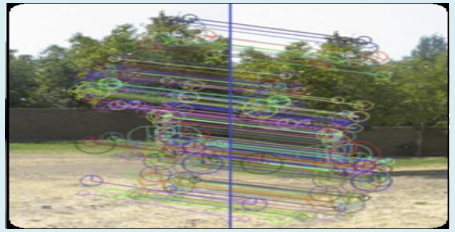
❖ Effect of the size of μ (step size or learning rate)

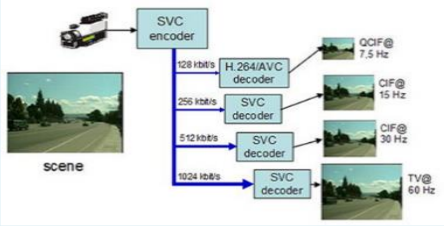


Large learning rate: Overshooting.



Small learning rate: Many iterations until convergence and trapping in local minima.





Contents

- GLOSSARY
- What is Neural Networks?
- Steepest Gradient Decent Algorithm
- **Regression Analysis**

❖ Regression analysis

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

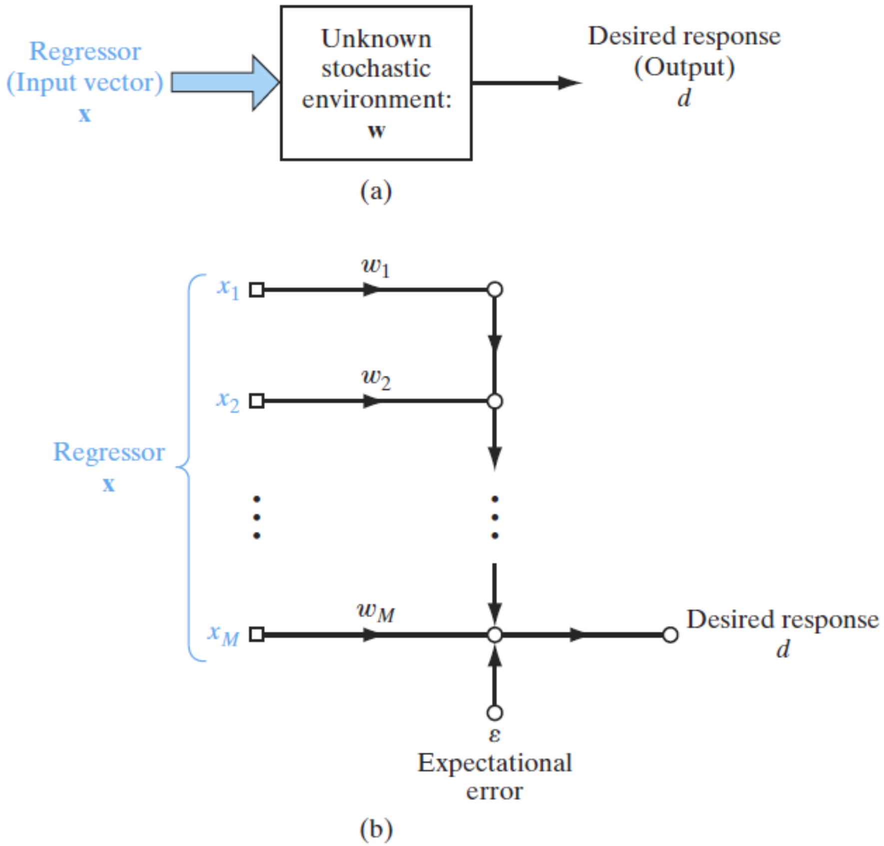
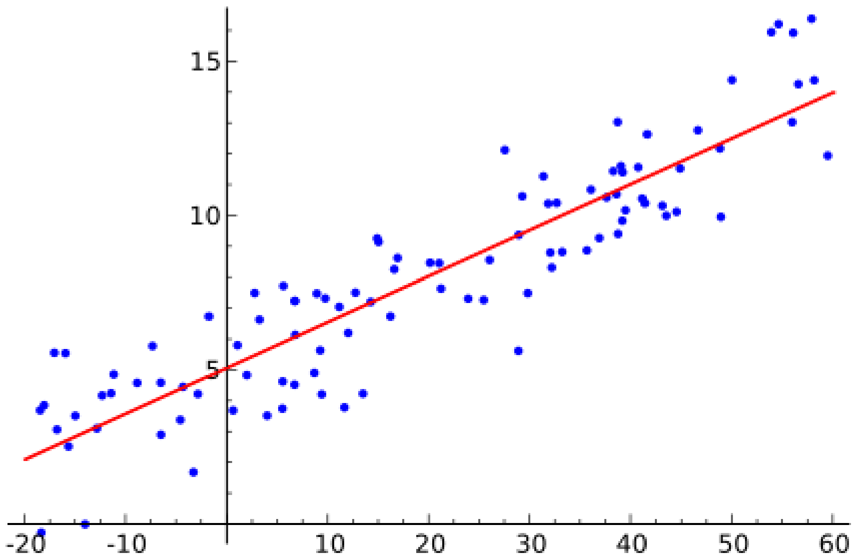
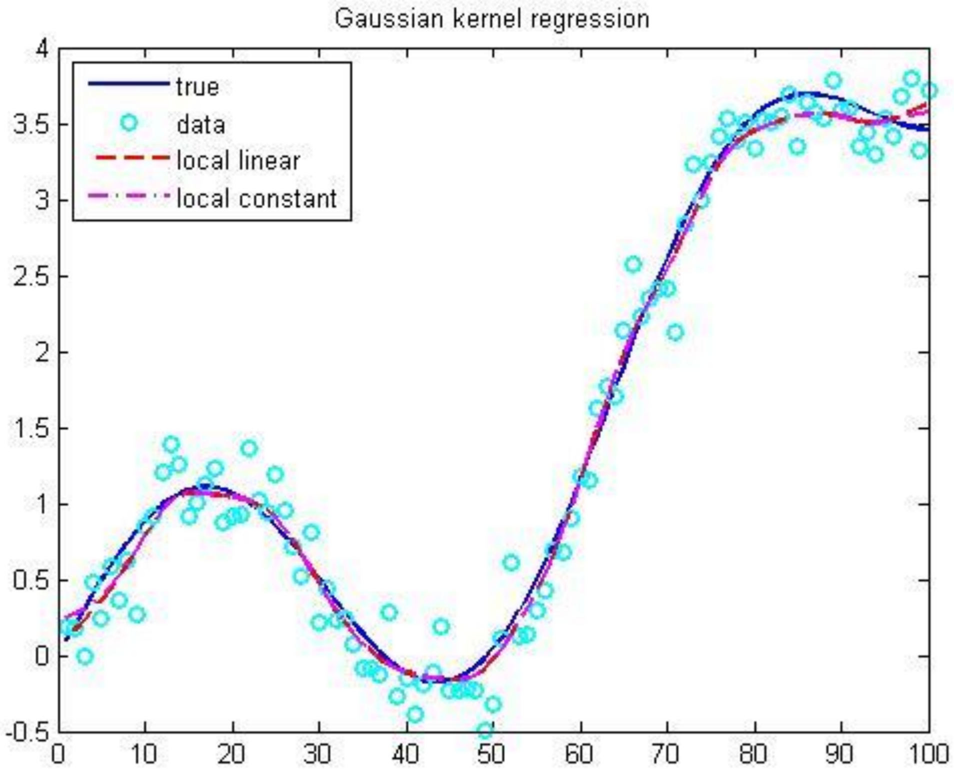


FIGURE 2.1 (a) Unknown stationary stochastic environment. (b) Linear regression model of the environment.

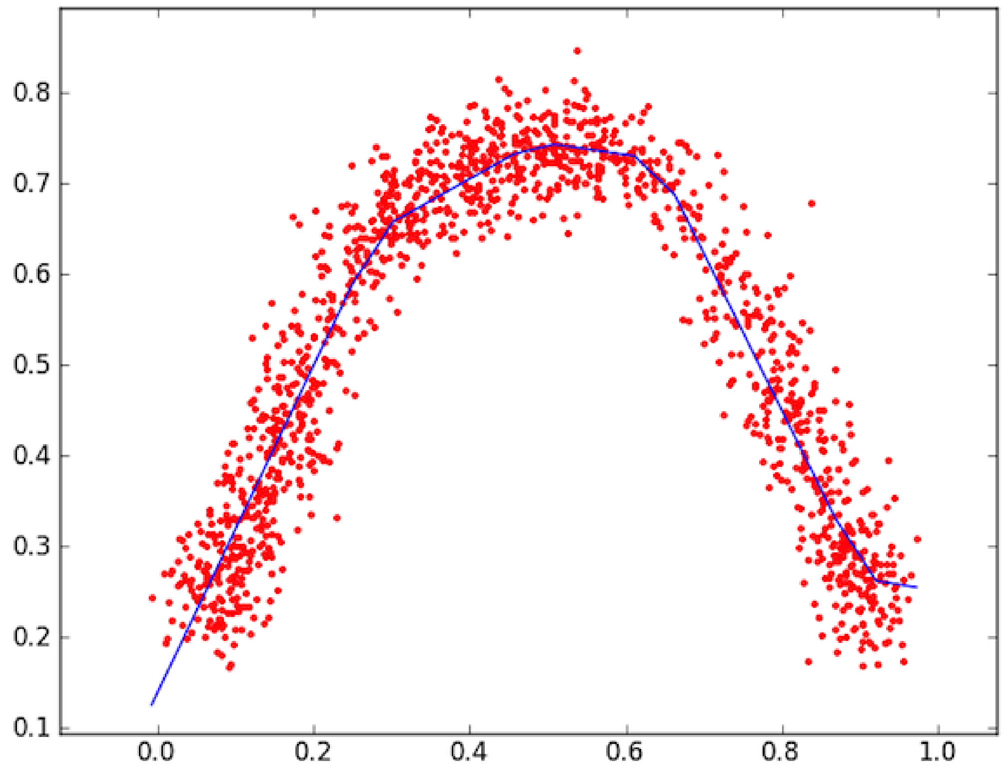
Given the joint statistics of the regressor ***X*** *and the corresponding response D, estimate the unknown parameter vector* **w**.

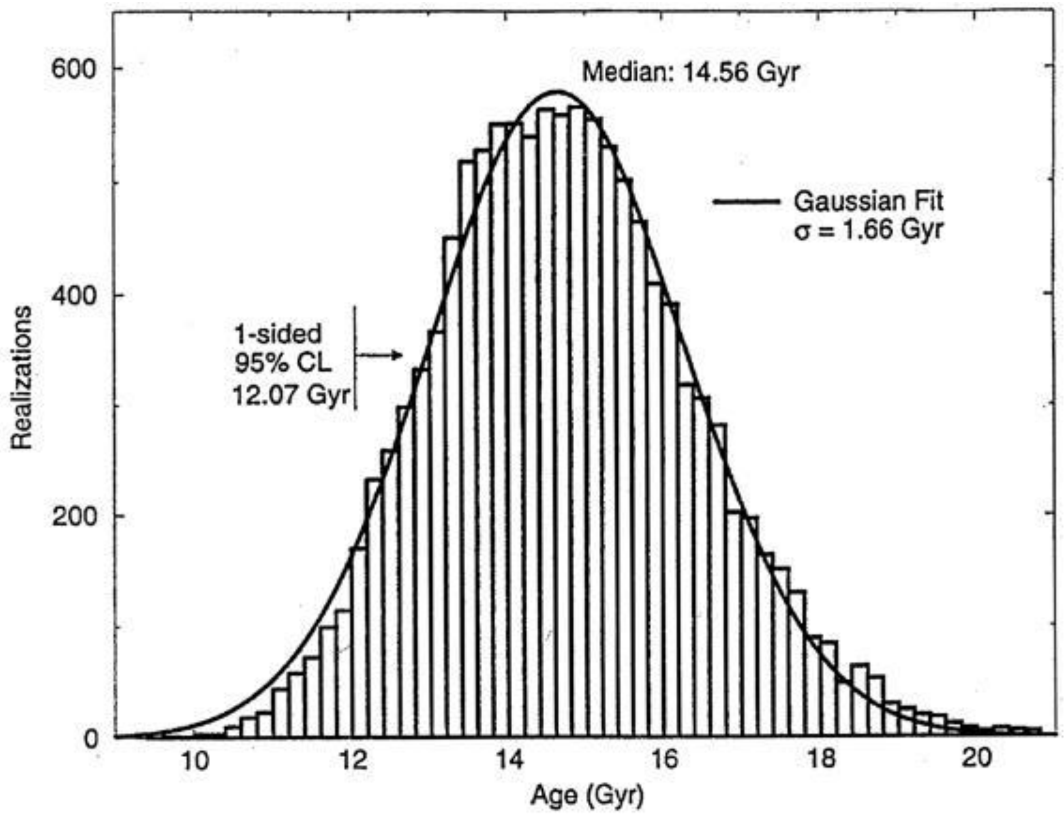
- **Data fitting (Curve fitting)** is a good example of regression analysis.





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





Thank you for your attention.!!!
QnA

https://ivpl.sookmyung.ac.kr