# EENG582
# Artificial Neural Networks

# Introduction

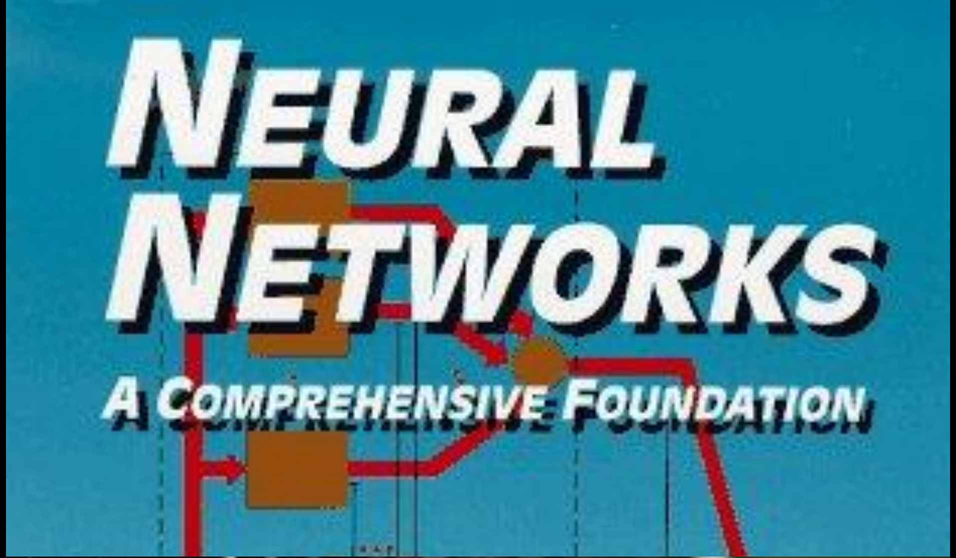
Sections 1.3 - 1.6– Model of a Neuron + NN Viewed as Directed Graphs + Neural Network Architecture

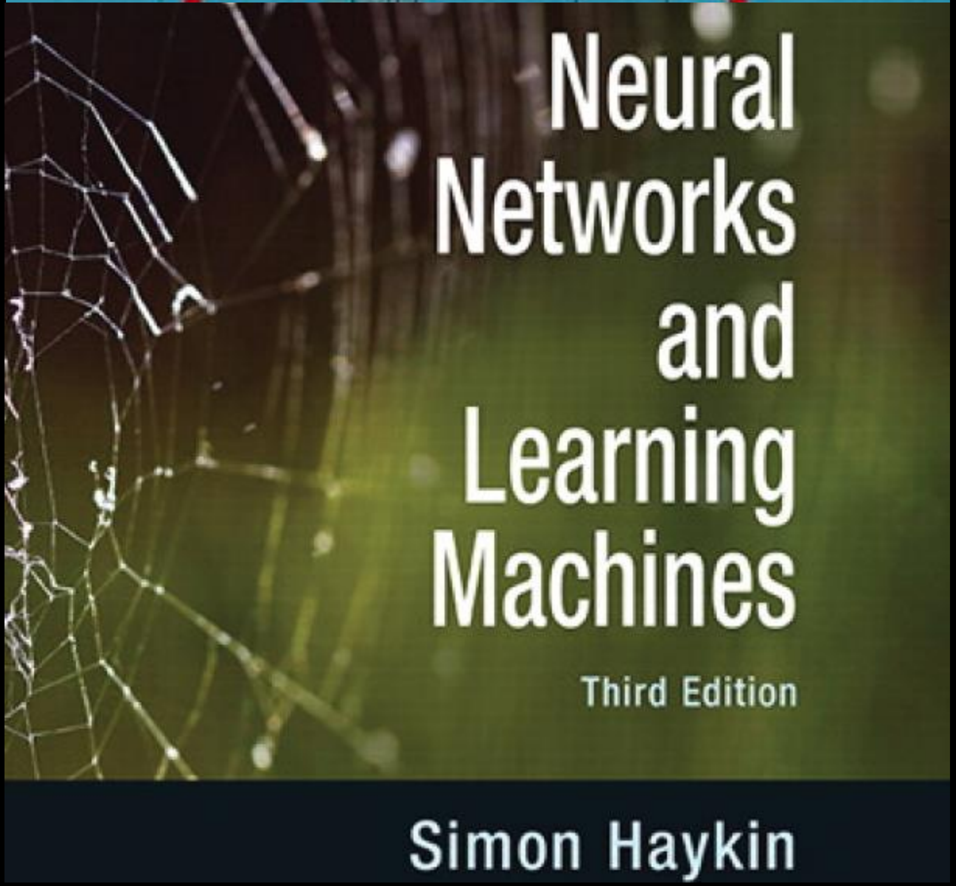
## Artificial Neural Networks

Prof. Dr. Hasan AMCA

Electrical and Electronic Engineering Department
(ee.emu.edu.tr)

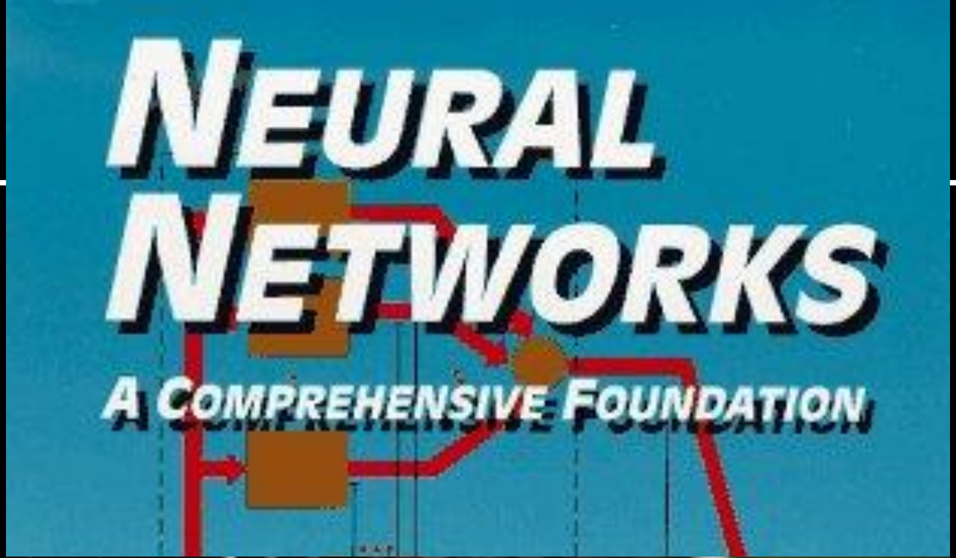
Eastern Mediterranean University
(emu.edu.tr)

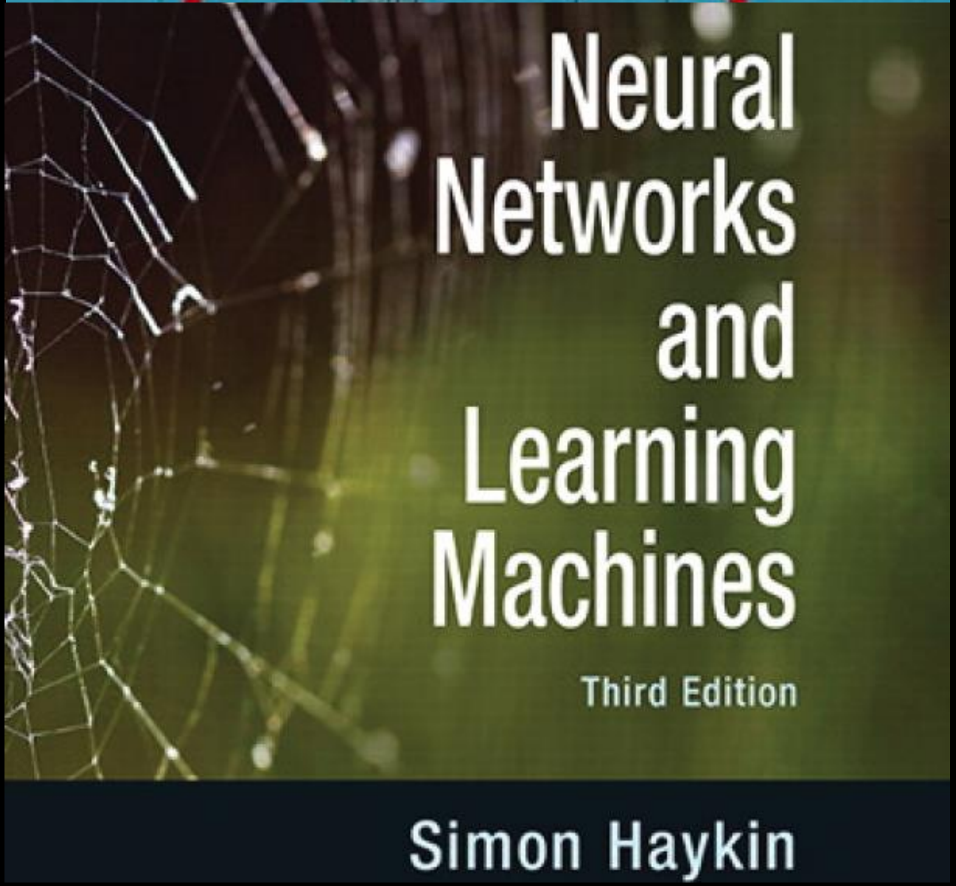


# 1 Introduction

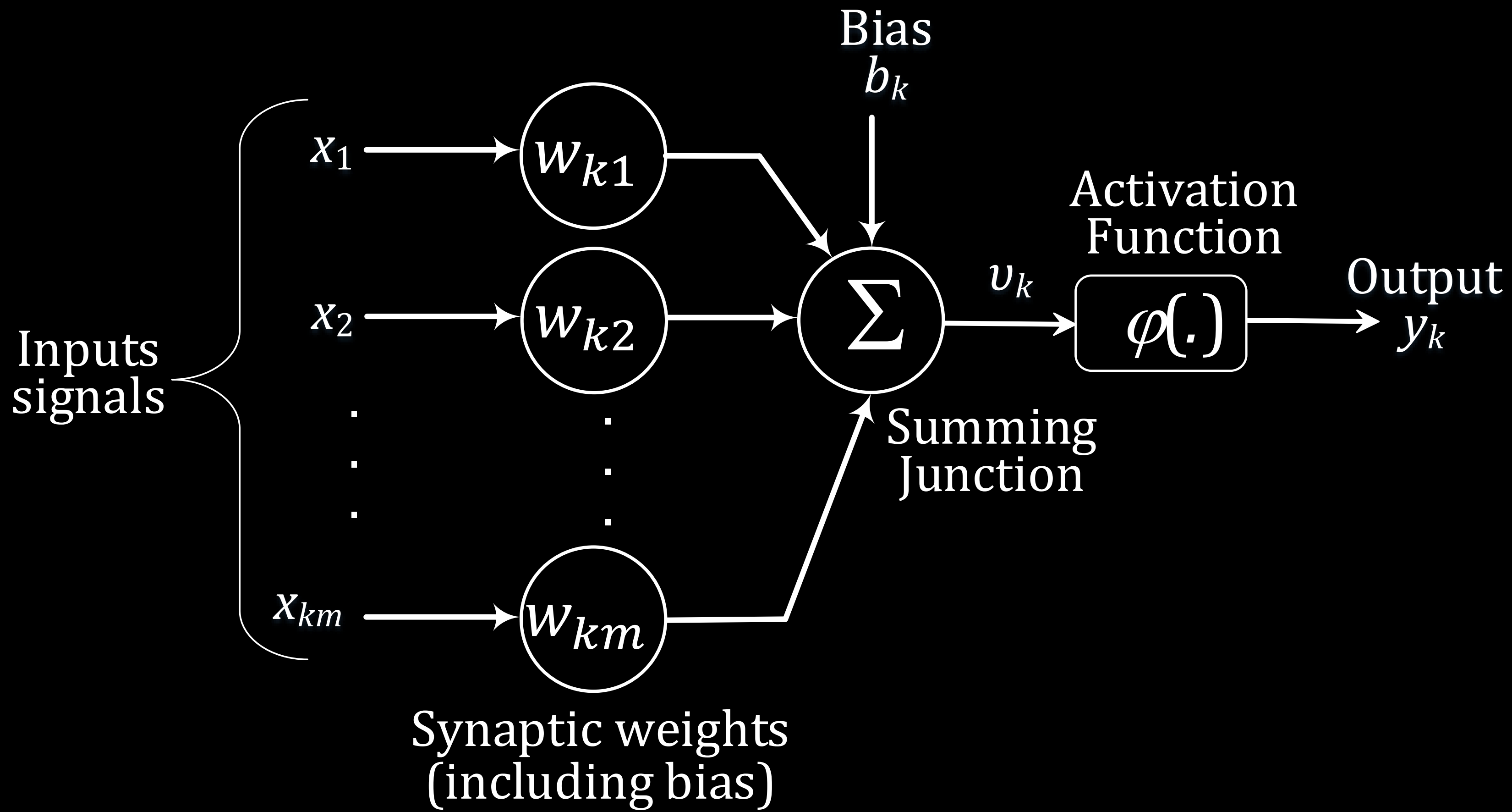
1.1 What Is a Neural Network?
1.2 Human Brain
1.3 Models of a Neuron
1.4 Neural Networks Viewed as Directed Graphs
1.5 Feedback
1.6 Network Architectures
1.7 Knowledge Representation
1.8 Artificial Intelligence and Neural Networks
1.9 Historical Notes
Notes and References
Problems



# 1.3 Models of a Neuron

- A neuron is an information-processing unit that is fundamental to the operation of a neural network. The block diagram of Fig. 1.5 shows the model of a neuron, which forms the basis for designing (artificial) neural networks

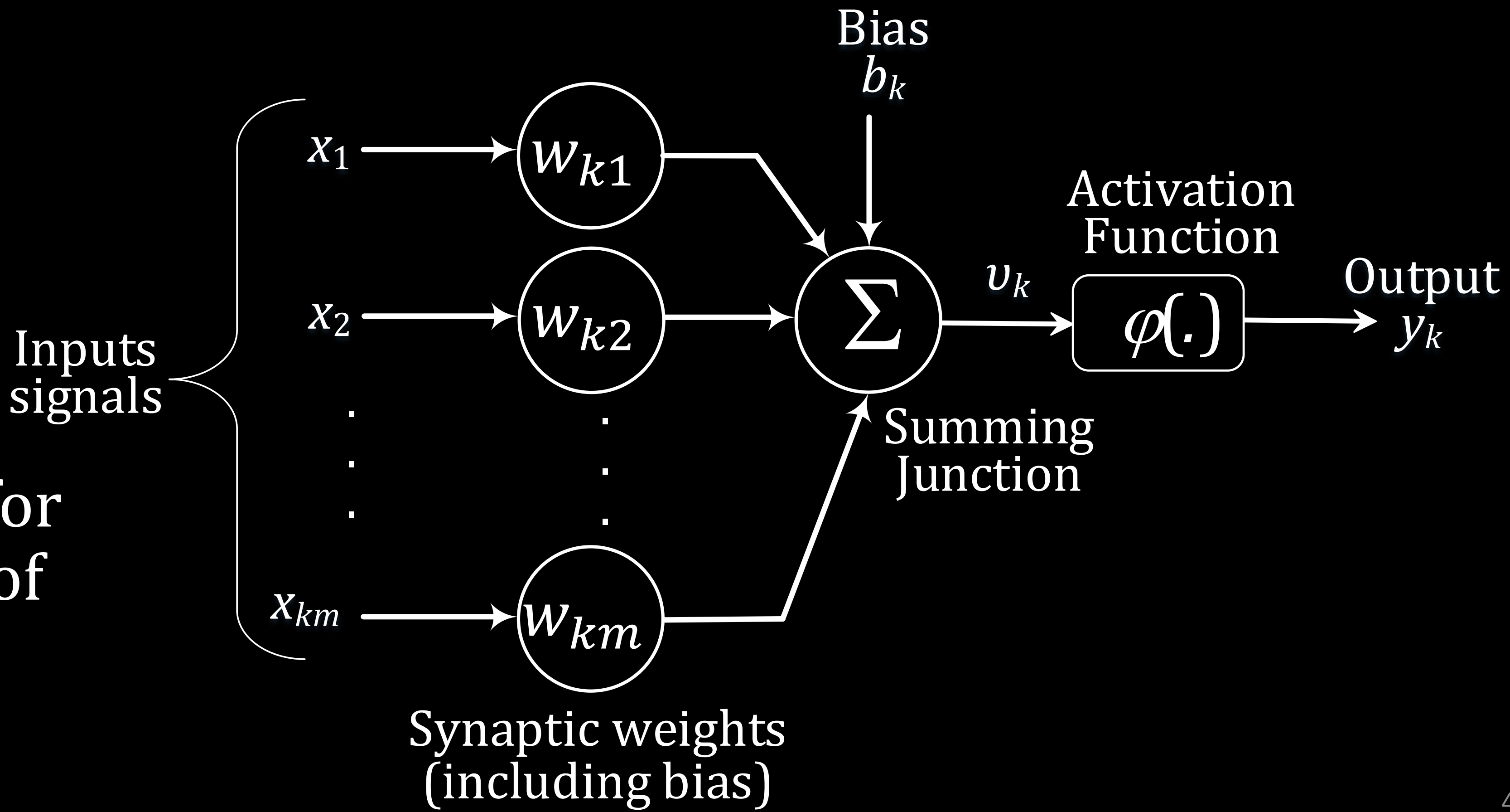


# Three Basic Elements of the Neuronal Model

1. A set of synapses or connecting links, each of which is characterized by a weight or strength of its own. Specifically, a signal $x_j$ at the input of synapse $j$ connected to neuron $k$ is multiplied by the synaptic weight $w_{kj}$

2. An adder for summing the input signals, weighted by the respective synapses of the neuron

3. An activation function for limiting the amplitude of the output of a neuron.



# Mathematical Description of a Neuron

- We may describe a neuron $k$ by writing the equations:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \qquad (1.1)$$

and

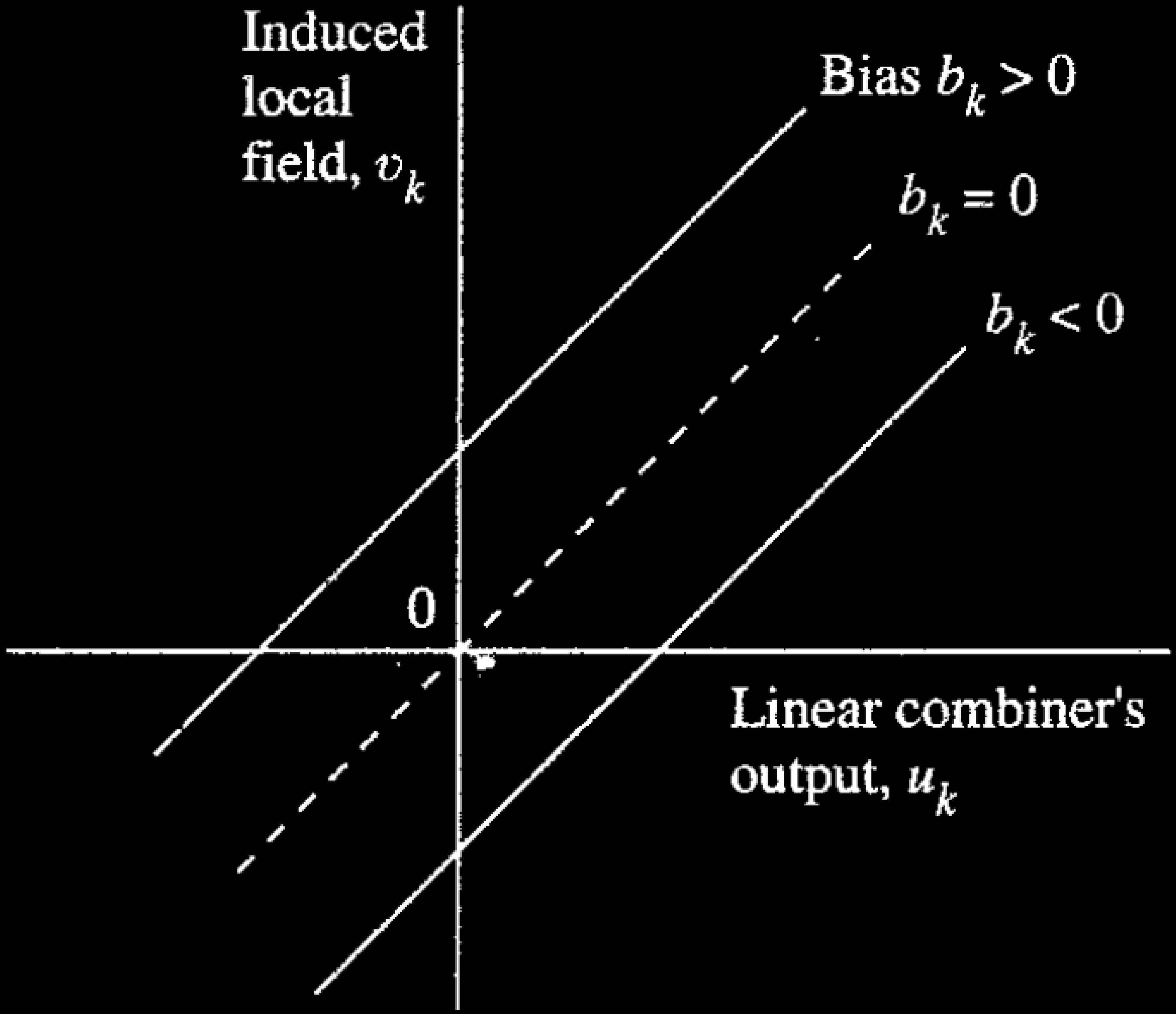
$$y_k = \varphi(u_k + b_k) \qquad (1.2)$$

- Where $x_1, x_2, x_3, \ldots\ldots x_m$ are the input signals and $w_{k1}, w_{k2}, \ldots w_{km}$ are the synaptic weights of neuron $k$; $u_k$ are the linear combiner output due to the input signals; $b_k$ is the bias; $\varphi(\,.\,)$ is the activation function; and $y_k$ is the output signal of the neuron.

- The use of bias $b_k$ applies an offline transformation to the output $u_k$

$$v_k = u_k + b_k \qquad (1.3)$$

# Fig. 1.6: Offline Transformation Produced by the Presence of a bias



# 1.2 Human Brain

- We may account for the bias $b_k$ as an external parameter of artificial neuron as

$$v_k = \sum_{j=0}^{m} w_{kj} x_j \tag{1.4}$$

and

$$y_k = \varphi(v_k) \tag{1.5}$$

we have added a new synapse

$$x_0 = +1 \tag{1.6}$$

and its weight is

$$w_{ko} = b_k \tag{1.7}$$

# Types of Activation Functions

- So, what non-linear activation functions can we choose?
- The model pigeon is a two-layer neural network
- Two different activation functions are used for the two layers
  - ~ The rectifier function (F.relu) is used at the outputs of the first layer and
  - ~ The sigmoid function (torch.sigmoid) for the outputs (yes, singular) of the second layer
  - ~ For example, activation1 in the Pigeon.forward method is a PyTorch tensor of dimension 30.

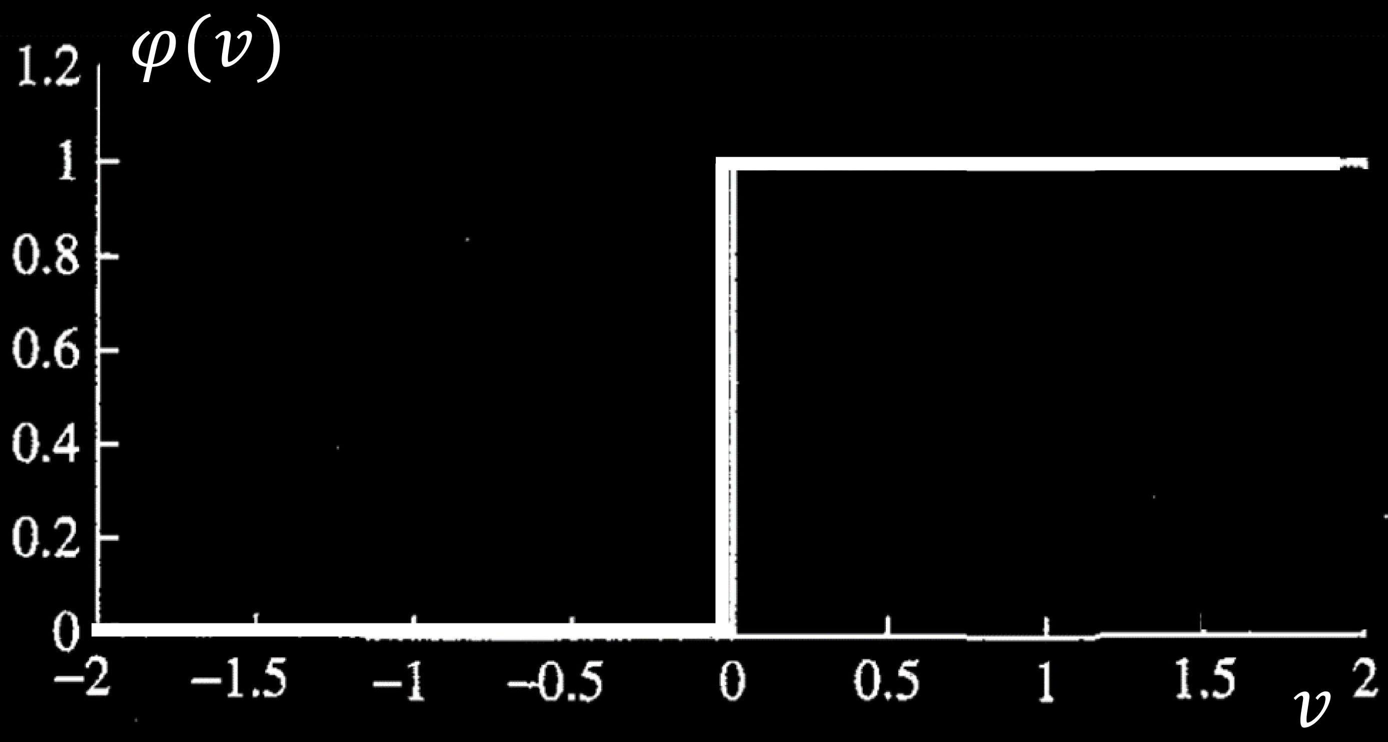


Source: https://www.cs.toronto.edu/~lczhang/360/lec/w02/terms.html

# The Threshold Function

- For the threshold type activation function, described in Fig. 1.8a, we have

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




- This is commonly referred to as a Heaviside function
- Correspondingly, the output of neuron $k$ employing such a threshold function is expressed as

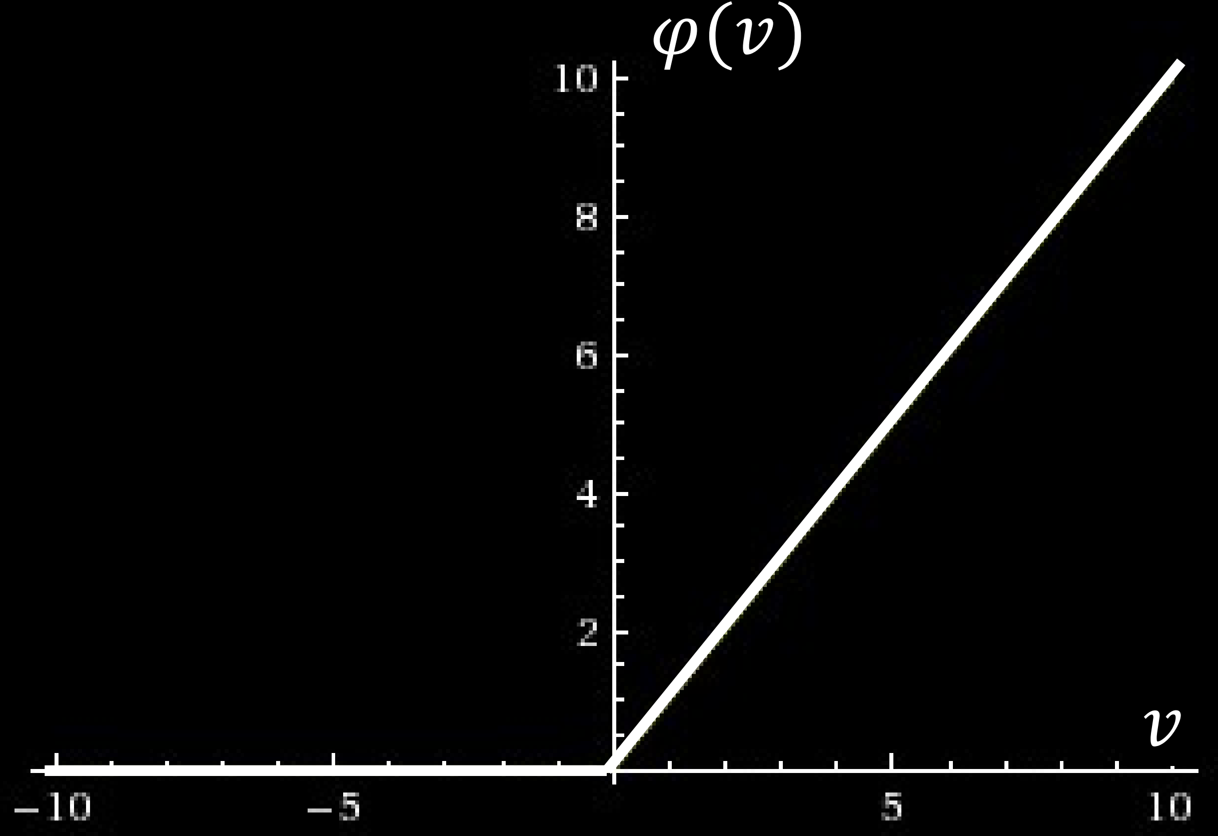
$$y_k = \begin{cases} 1 \ \text{if} \ v_k \geq 0 \\ 0 \ \text{if} \ v_k < 0 \end{cases}$$

- where $v_k$ is the induced local field of the neuron; that is,

$$v_k = \sum_{j=1}^{m} w_{kj} x_j + b_k$$

# Rectifier Function

- A linear rectifier function has the i/p – o/p characteristics shown in the figure
- The function is linear for activation above zero  and equal to zero otherwise
- An artificial neuron unit that uses the rectifier function as its non-linearity is called a rectified linear unit (ReLU)
- Most machine learning practitioners use ReLU units for intermediate layers of a neural network
- Due to the simple mathematics, the ReLU unit, networks with ReLU units are easier to optimize than those using sigmoid activation.

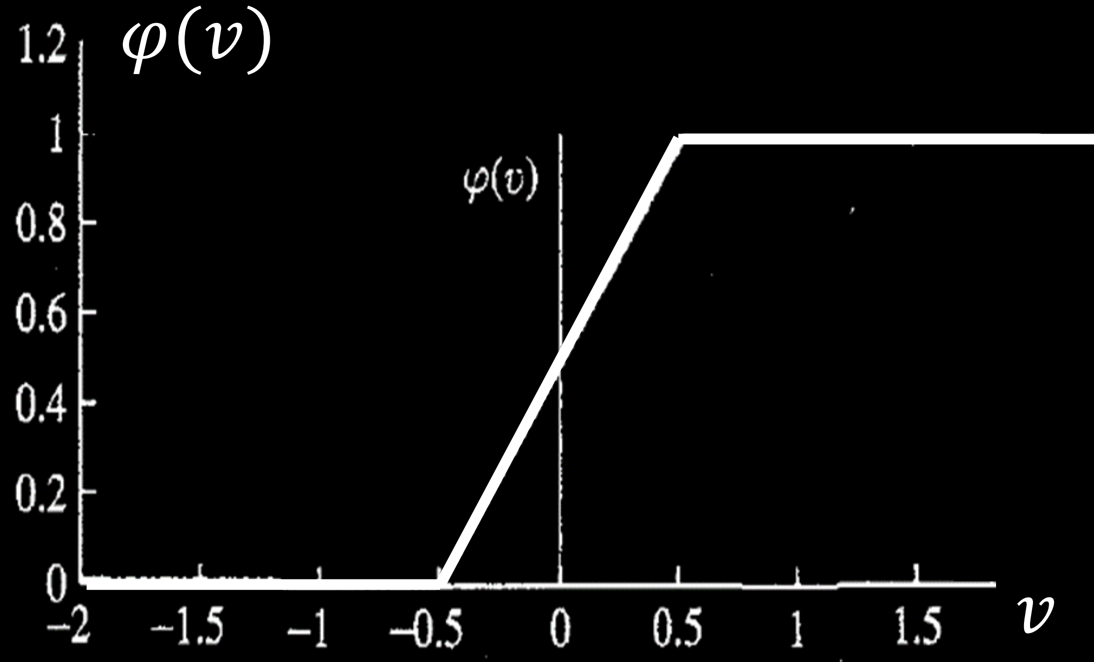



Source: https://www.cs.toronto.edu/~lczhang/360/lec/w02/terms.html

# Piecewise Linear Function

- For the piecewise-linear function shown in the figure, we have

$$\varphi(v) = \begin{cases} 1 & v \geq 1/2 \\ v & +\frac{1}{2} > v > -1/2 \\ 0 & v \leq -1/2 \end{cases}$$




Here, the amplification factor inside the linear region of operation is unity.

- The following 2 cases are viewed as special forms of piecewise-linear function
  - ~ A linear combiner arises if the linear region of operation is maintained without running into saturation.
  - ~ The piecewise-linear function reduces to a threshold function if the amplification factor of the linear region is made infinitely large.

# Sigmoid Function

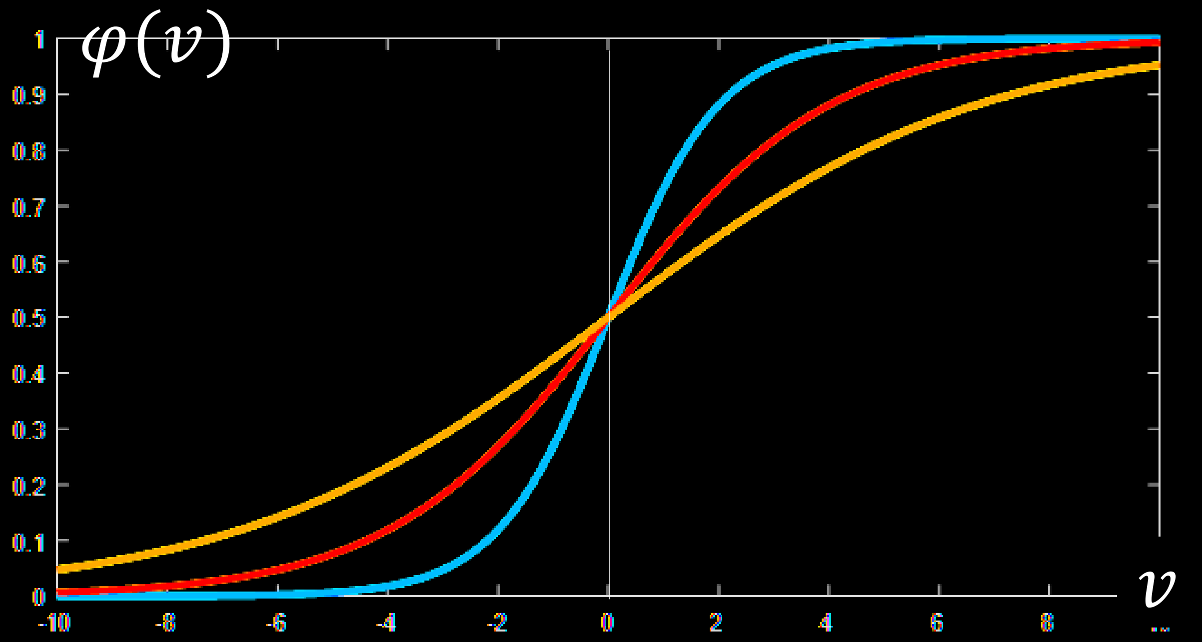
- Sigmoid function, is the most common form of activation function used in construction of artificial neural networks. It is defined as a strictly increasing function, exhibiting a graceful balance between linear and nonlinear behavior

The sigmoid function is defined by

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

where $a$ is the slope parameter

By varying the parameter $a$ , we obtain sigmoid functions of different slopes, as illustrated in Fig. 1.8c.

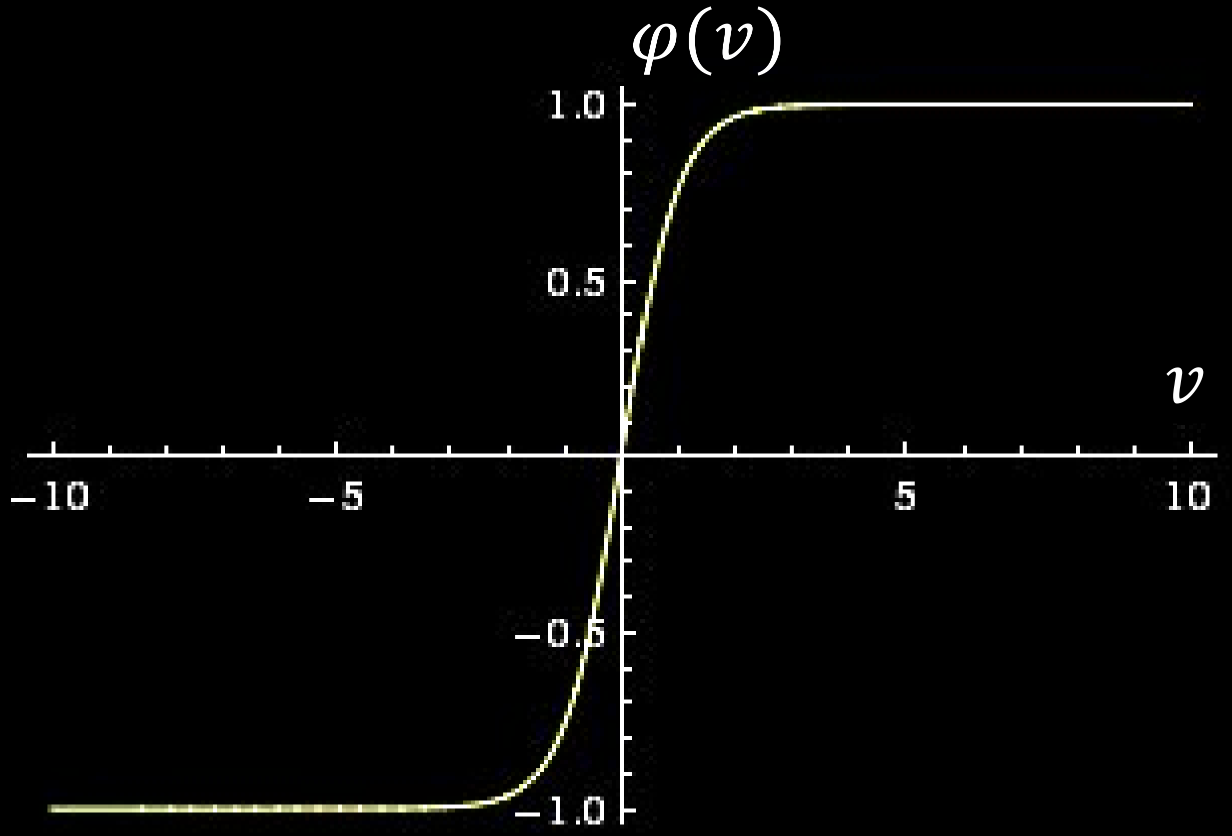



- In fact, the slope at the origin equals $a/4$. In the limit, as the slope parameter approaches infinity, the sigmoid function becomes a threshold function.
- Note that sigmoid function is differentiable, whereas threshold function is not.

# Tanh Function

- The tanh function is a variation of the sigmoid function.
- The output of the tanh function is always between -1 and 1 (instead of 0 and 1)

$$\varphi(v) = \tanh(v) \qquad (1.14)$$




Source: https://www.cs.toronto.edu/~lczhang/360/lec/w02/terms.html

# Stochastic Model of a Neuron

- The neuronal model described in Fig.1.7 is deterministic in that its input-output behavior is precisely defined for all inputs
- The decision for a neuron to fire (i.e., switch its state from "off" to "on") is probabilistic where a neuron is permitted to reside in only one of two states: + 1 or -1
- Let $x$ denote the state of the neuron, and $P(v)$ denote the probability of firing, where $v$ is the induced local field of the neuron. We may then write

$$x = \begin{cases} +1 & \text{with probability } P(v) \\ -1 & \text{with probability } 1 - P(v) \end{cases}$$

- .. A standard choice for $P(v)$ is the sigmoid-shaped function

$$P(v) = \frac{1}{1+\exp(-v/T)} \qquad (1.15)$$

- where $T$ is a *pseudo-temperature* used to control the noise level

# 1.4 Neural Networks Viewed as Directed Graphs

- The block diagram of Fig. 1.5 or that of Fig. 1. 7 provides a functional description of the elements that constitute an artificial neuron model
- We may simplify the appearance of the model by using the signal-flow graphs with a well-defined set of rules

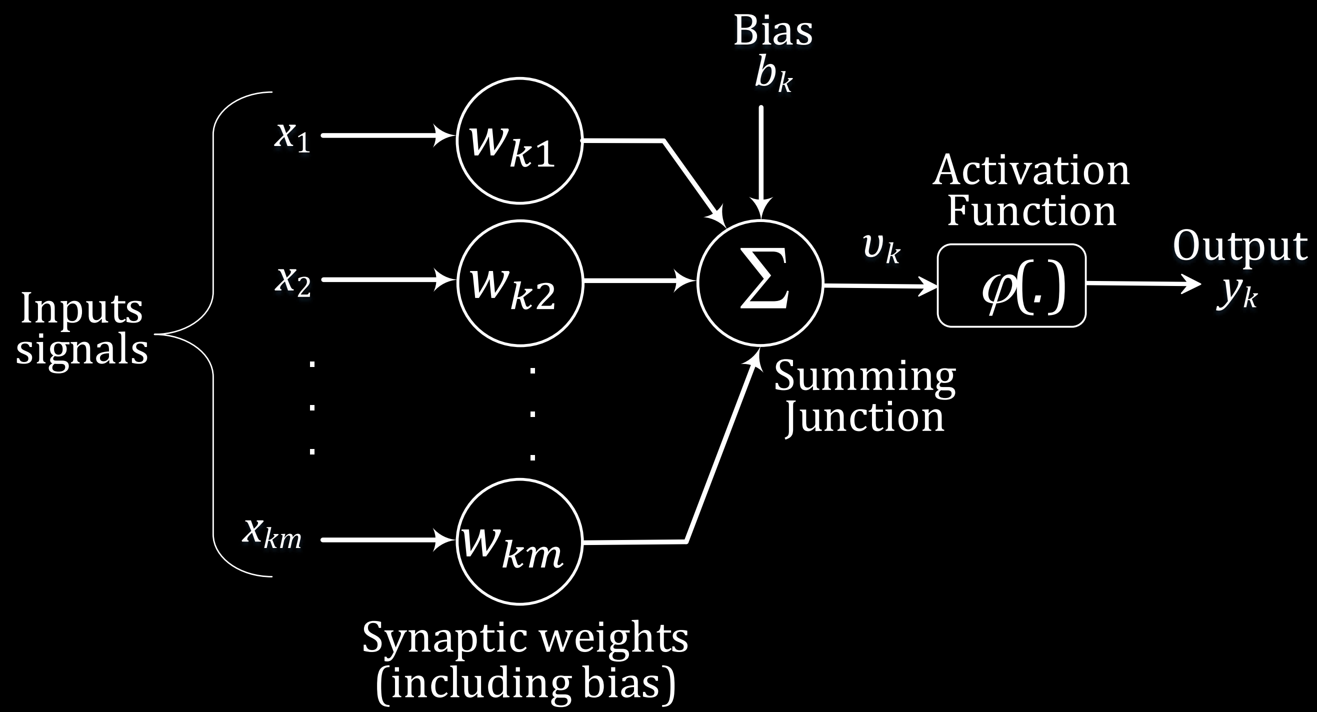



Fig. 1.5

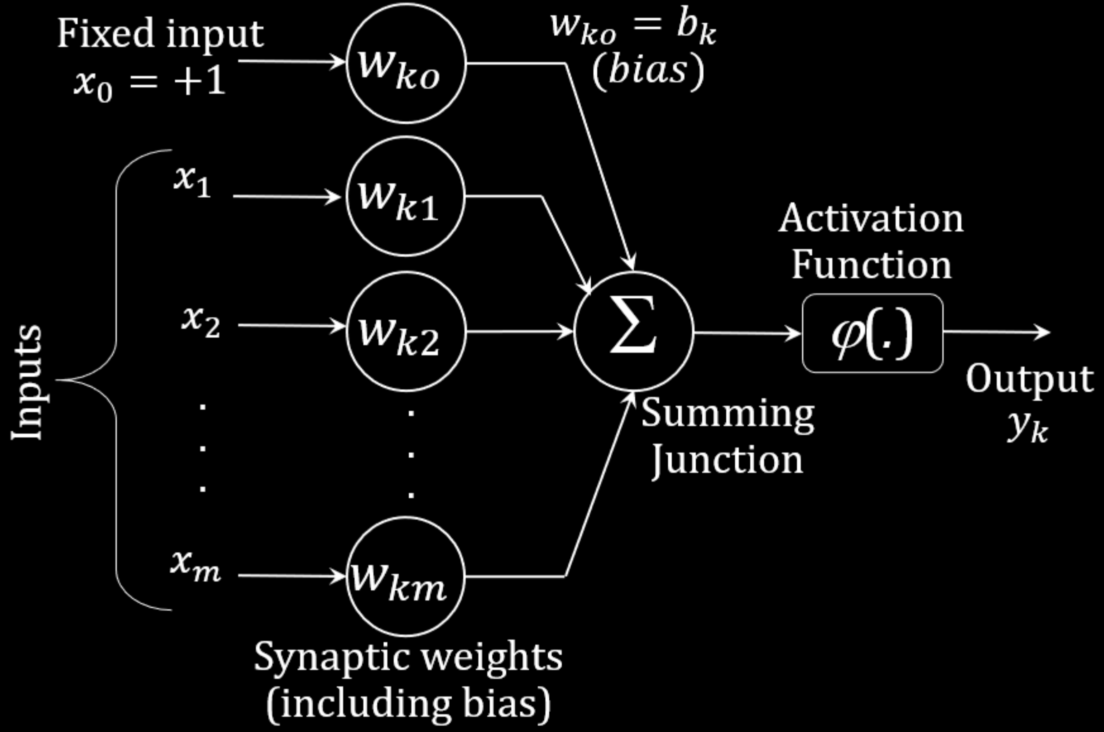
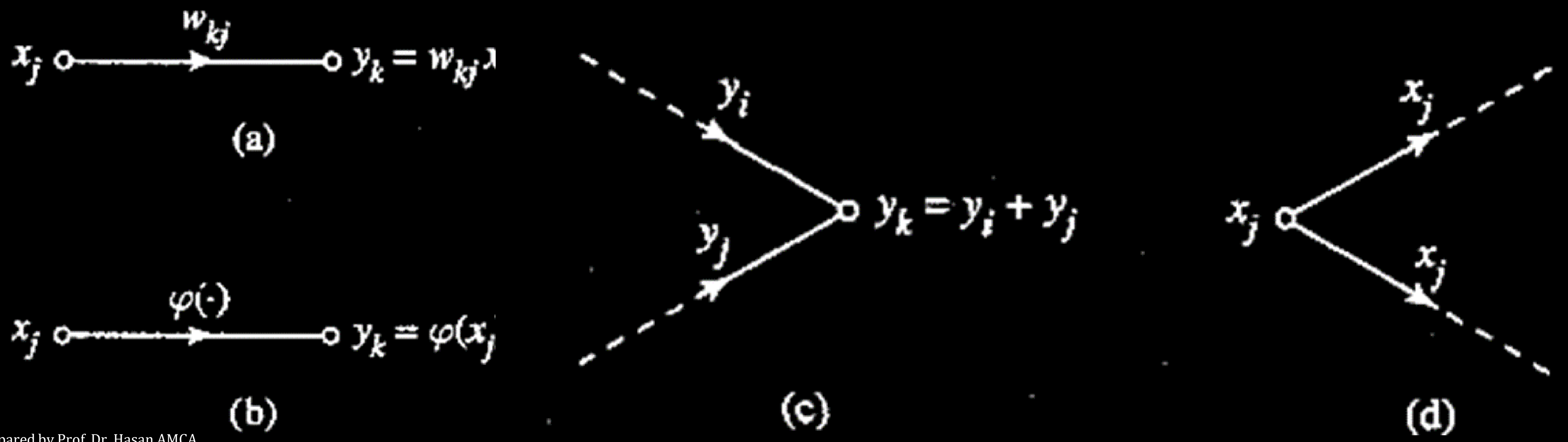



Fig. 1.7

# 1.4 Neural Networks Viewed as Directed Graphs

- A signal-flow graph is a network of directed links (branches) that are interconnected at certain points called nodes
- A typical node $j$ has an associated node signal $x_j$ where a directed link originates at node $j$ and terminates on node $k$
- it has an associated transfer function or transmittance that specifies the manner in which the signal $y_k$ at node $k$ depends on the signal $x_j$ at node $j$.



# 1.4 Neural Networks Viewed as Directed Graphs

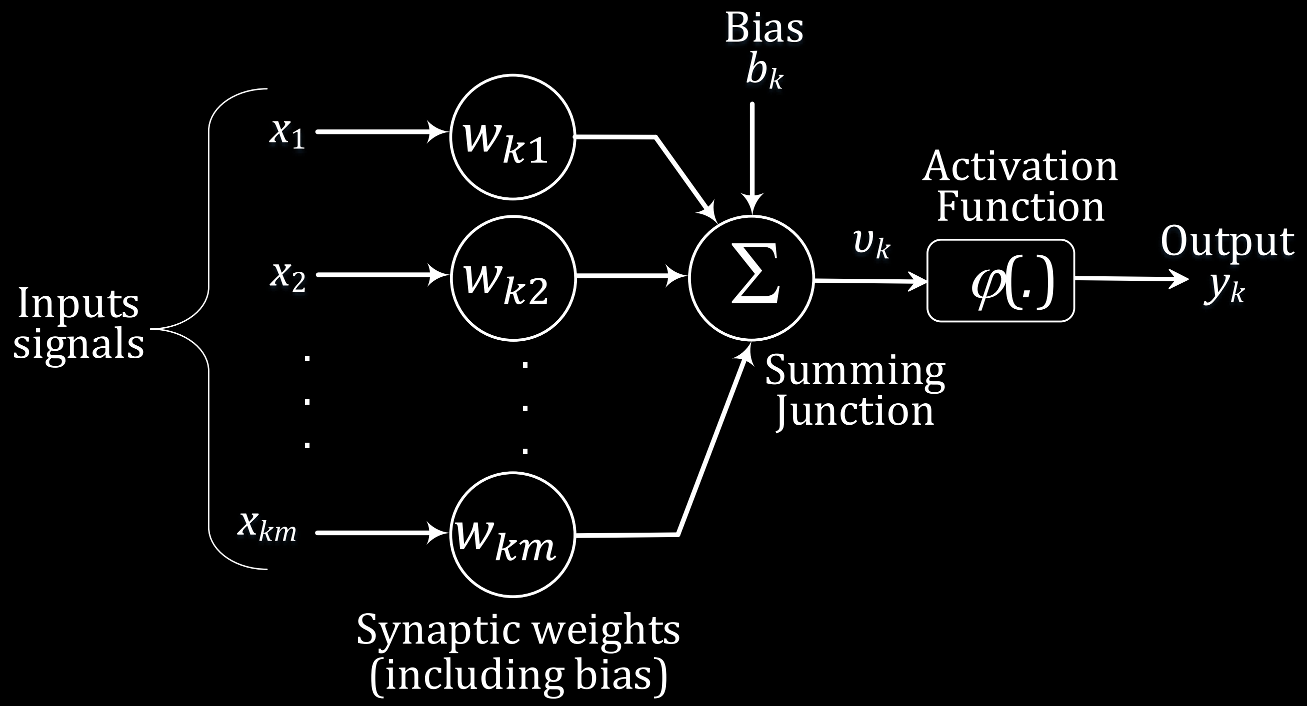



Fig. 1.5

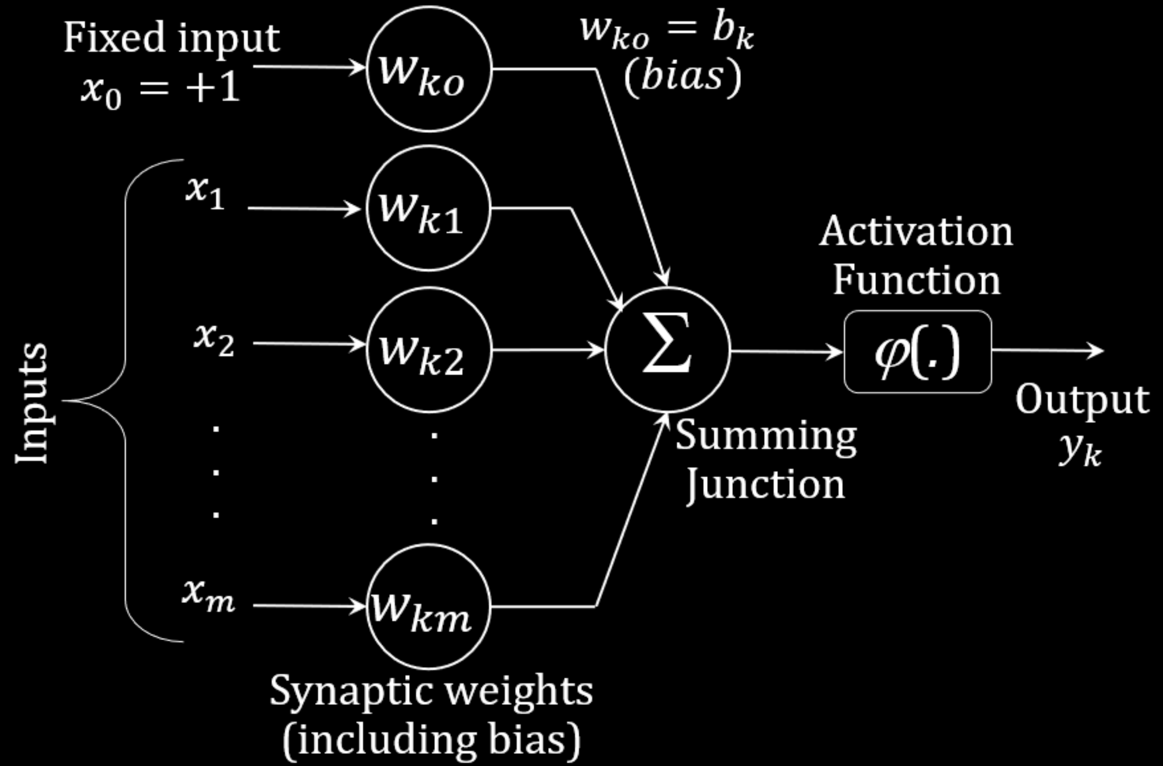
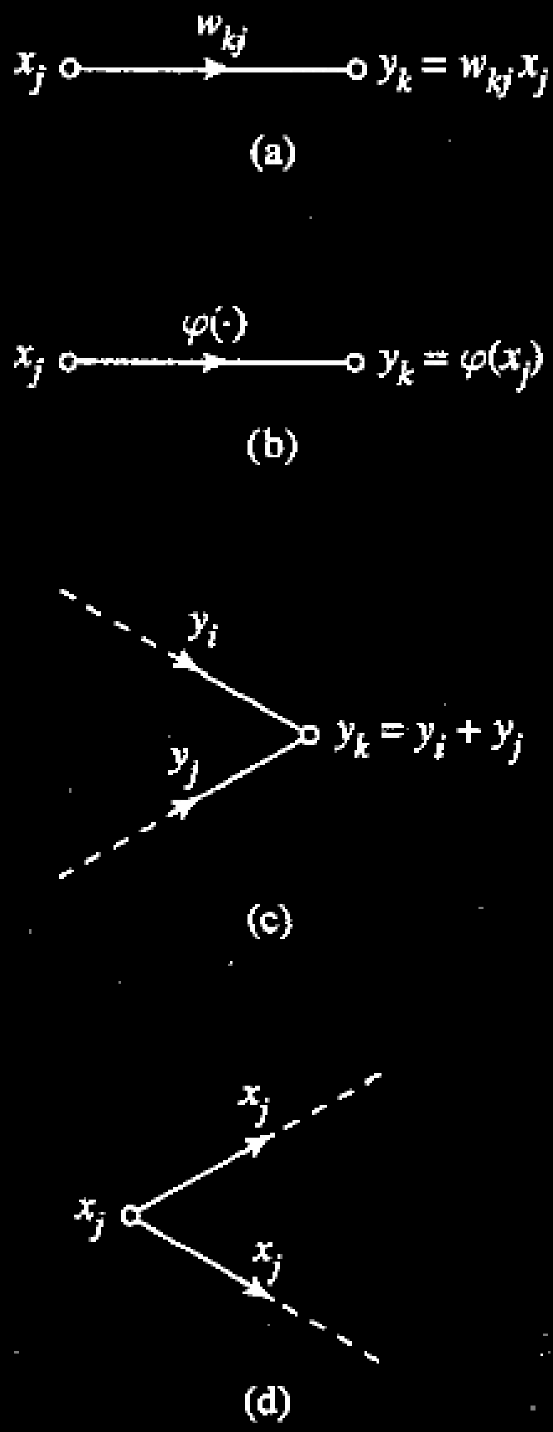



Fig. 1.7

# Signal-Flow Graph Rules

- **Rule 1:** A signal flows along a link only in the direction defined by the arrow on the link. There are two different types of links
  a) Synaptic links, whose behavior is governed by a linear input-output relation. Specifically, the node signal $x_j$ is multiplied by the synaptic weight $w_{kj}$ to produce the node signal $y_k$ as illustrated in Fig. 1.9a.
  b) Activation links, whose behavior is governed in general by a nonlinear input-output relation.
- **Rule 2:** A node signal equals the algebraic sum of all signals entering the pertinent node via the incoming links.
- **Rule 3:** The signal at a node is transmitted to each outgoing link originating from that node, with the transmission being entirely independent of transfer functions of outgoing links



# 1.4 Neural Networks Viewed as Directed Graphs

- The block diagram of Fig. 1.5 or that of Fig. 1. 7 provides a functional description of the elements that constitute an artificial neuron model
- We may simplify the appearance of the model by using the signal-flow graphs with a well-defined set of rules

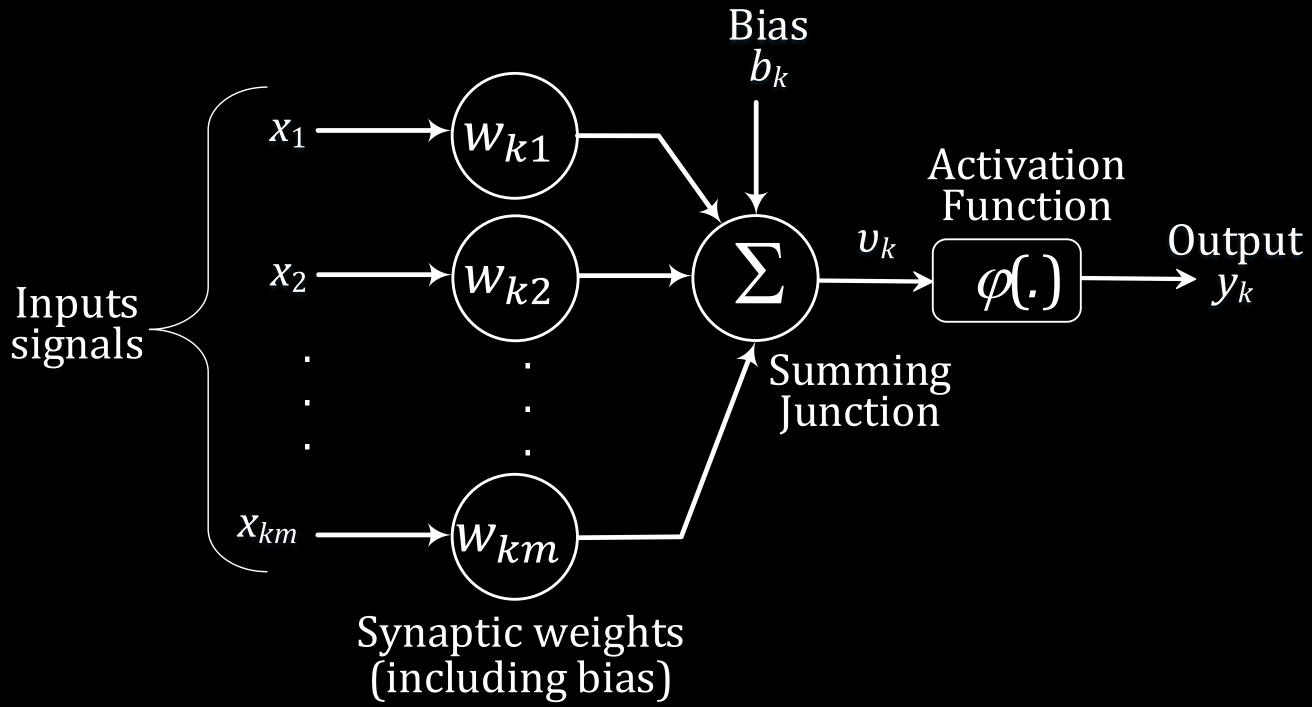



Fig. 1.5

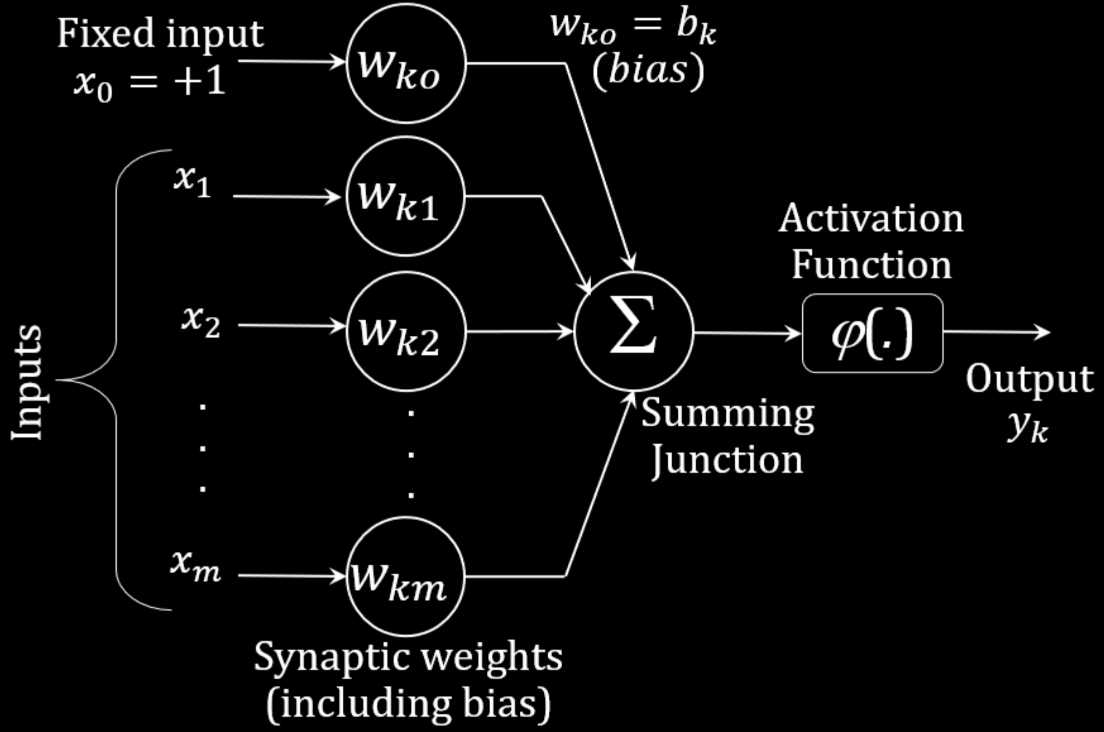
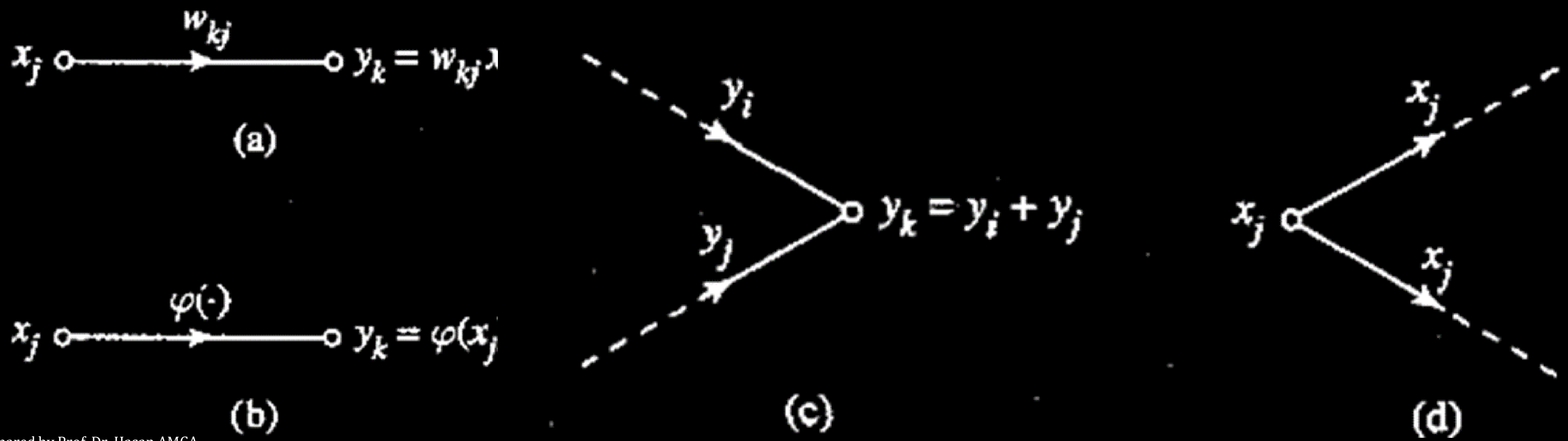



Fig. 1.7

# 1.4 Neural Networks Viewed as Directed Graphs

- A signal-flow graph is a network of directed links (branches) that are interconnected at certain points called nodes
- A typical node *j* has an associated node signal $x_j$ where a directed link originates at node *j* and terminates on node *k*
- it has an associated transfer function or transmittance that specifies the manner in which the signal $y_k$ at node *k* depends on the signal $x_j$ at node *j*.



# 1.4 Neural Networks Viewed as Directed Graphs

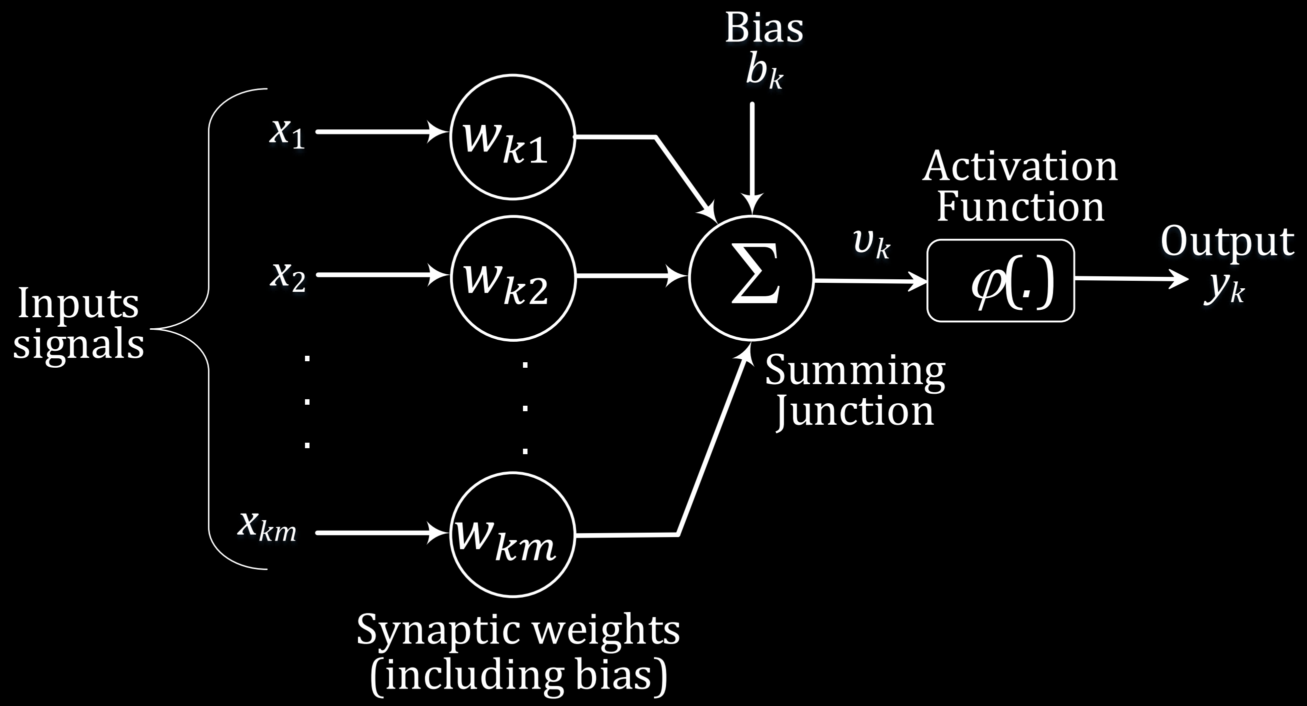



Fig. 1.5

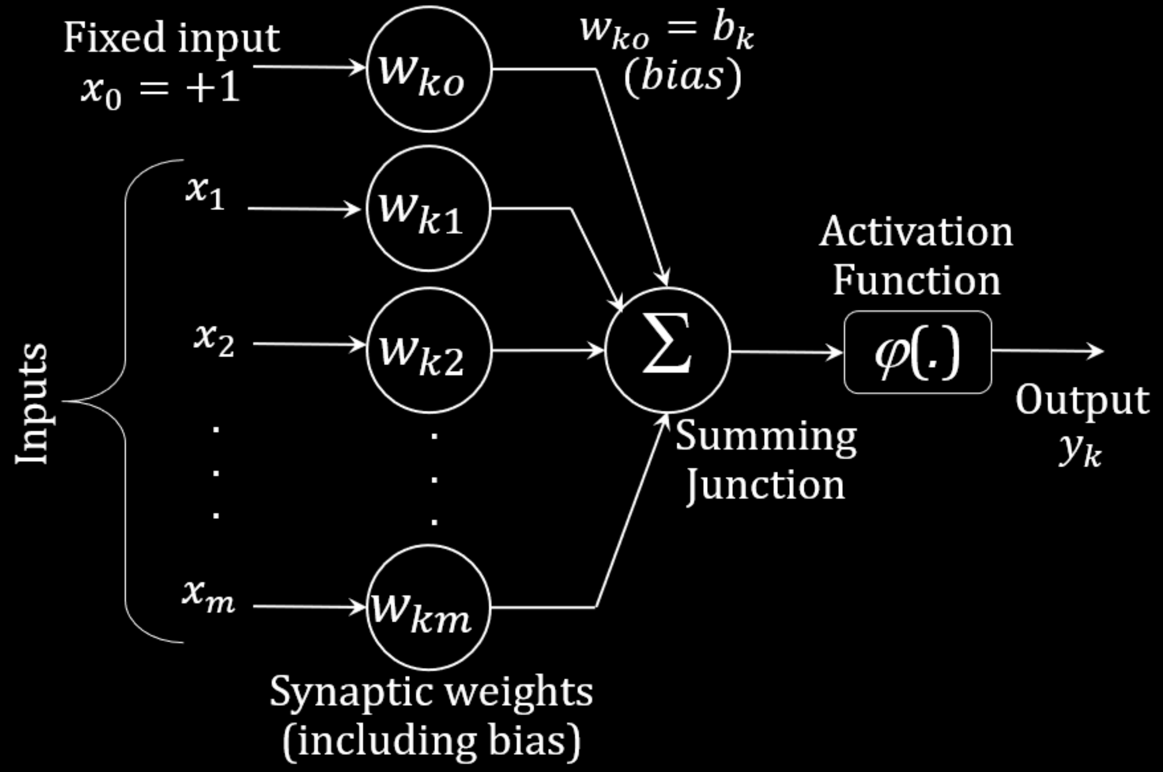



Fig. 1.7

# Signal-Flow Graph Rules

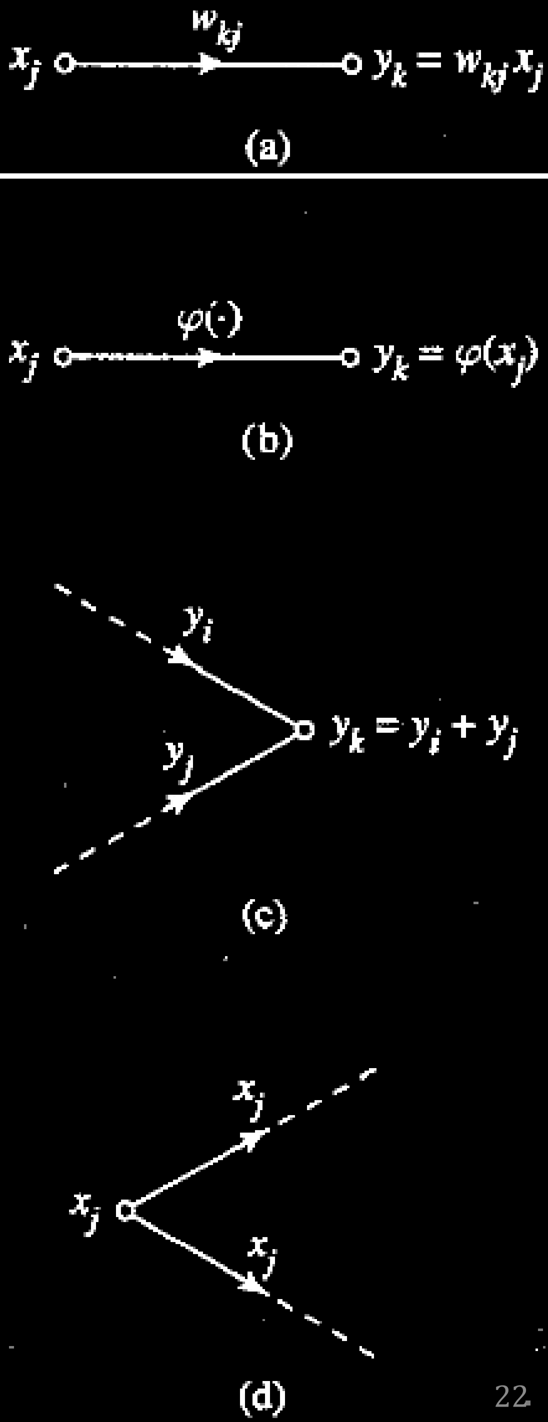
- **Rule 1:** A signal flows along a link only in the direction defined by the arrow on the link. There are two different types of links

  a) Synaptic links, whose behavior is governed by a linear input-output relation. Specifically, the node signal $x_j$ is multiplied by the synaptic weight $w_{kj}$ to produce the node signal $y_k$ as illustrated in Fig. 1.9a.

  b) Activation links, whose behavior is governed in general by a nonlinear input-output relation.

- **Rule 2:** A node signal equals the algebraic sum of all signals entering the pertinent node via the incoming links.

- **Rule 3:** The signal at a node is transmitted to each outgoing link originating from that node, with the transmission being entirely independent of transfer functions of outgoing links



# 1.5 Feedback

- Feedback exists in a dynamic system whenever the output of an element in the system influences the input, thereby giving rise to closed paths for the transmission of signals around the system
- Feedback plays a major role in the study of neural networks known as recurrent networks
- Figure 1.12 shows the signal-flow graph of a single-loop feedback system,
- The feed forward path and the feedback path that are characterized by the "operators" A and B, respectively.
- From Fig 1.12 we readily note the following input-output relationships:

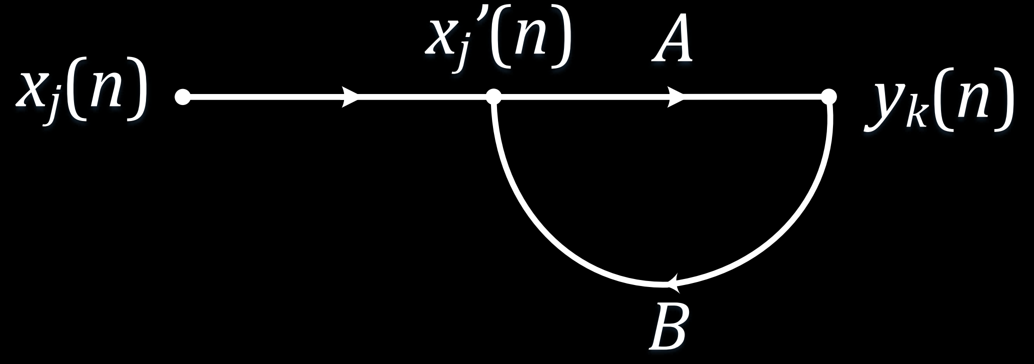



Fig. 1.12 . Signal-flow graph of a single-loop feedback system.

# Input-output Relationship of a Neuron

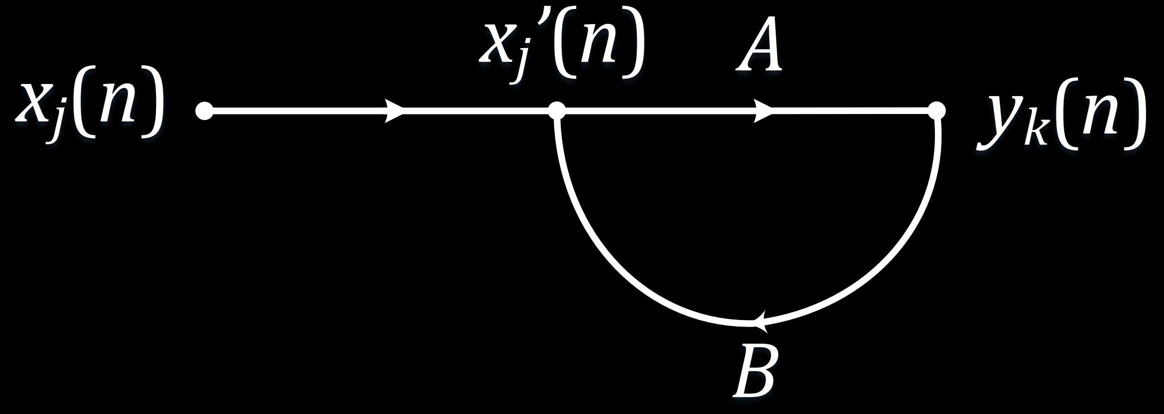
- The following input-output relationship of a neuron exists

$$y_k(n) = A\left[x_j'(n)\right] \qquad (1.16)$$

$$x_j'(n) = x_j(n) + B[y_k(n)] \qquad (1.17)$$

- where A and B are operators.
- Eliminating $x_j'(n)$ between Eq. (1.16) and (1.17), we get

$$y_k(n) = \frac{A}{1+AB}[x_j(n)] \qquad (1.18)$$



# Input-output Relationship of a Neuron

- For the single-loop feedback system shown in Fig. 1.13, we may express the closed loop operator of the system as

$$\frac{A}{1 - AB} = \frac{w}{1 - wz^{-1}} = w(1 - wz^{-1})^{-1}$$

- Using the binomial expansion for $(1 - wz^{-1})^{-1}$, we may rewrite the closed-loop operator of the system as

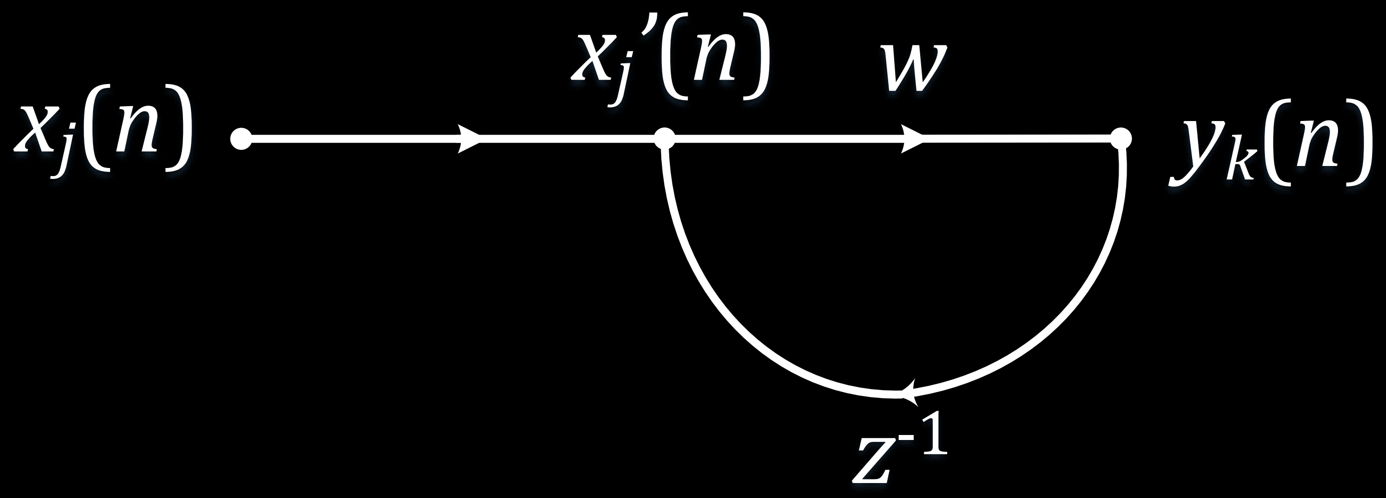



Fig. 1.13 Signal-flow graph of a first-order, infinite-duration impulse response (IIR) filter.

# Input-output Relationship of a Neuron

- The following input-output relationship of a neuron exists

$$y_k(n) = A\left[x_j'(n)\right] \qquad (1.16)$$

$$x_j'(n) = x_j(n) + B[y_k(n)] \qquad (1.17)$$

- where A and B are operators.
- Eliminating $x_j'(n)$ between Eqs. (1.16) and (1.17), we get

$$y_k(n) = \frac{A}{1+AB}[x_j(n)] \qquad (1.18)$$

- For the single-loop feedback system shown in Fig. 1.13, we may express the closed loop operator of the system as

$$\frac{A}{1-AB} = \frac{w}{1-wz^{-1}} = w(1-wz^{-1})^{-1}$$

# 1.5 Feedback

$$\frac{A}{1-AB} = w\sum_{l=0}^{\infty} w^l z^{-l} \qquad (1.19)$$

- Hence, substituting Eq. (1.19) in (1.18), we get

$$y_k(n) = w\sum_{l=0}^{\infty} w^l z^{-l}\left[x_j(n)\right] \qquad (1.20)$$

from the definition of $z^{-1}$ we have

$$z^{-l}\left[x_j(n)\right] = x_j(n-l) \qquad (1.21)$$

and the output signal is

$$y_k(n) = \sum_{l=0}^{\infty} w^{l+1} x_j(n-l) \qquad (1.22)$$

The 2 different cases of intertest are stable and unstable described as follows:

1. When $|w| < 1$, the output signal $y_k(n)$ is exponentially convergent and the system is stable. This is illustrated in Fig. 1.14a for a positive $w$.
2. When $|w| \geq 1$, the output signal $y_k(n)$ is divergent and the system is unstable as shown in Fig. 1.14b and Fig 1.14c

# 1.5 Feedback

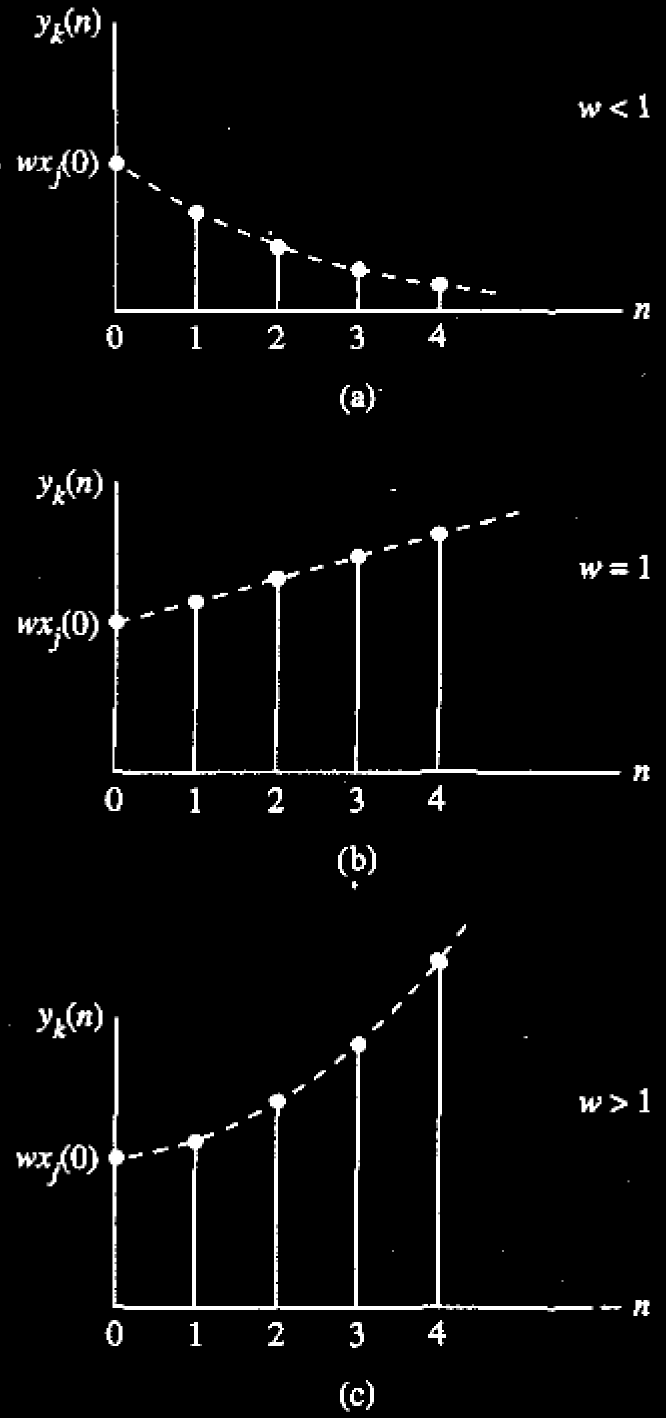



**Fig. 1.14.** Time response of Fig. 1.13 for three different values of forward weight *w.*

(a) Stable

(b) Linear divergence

(c) Exponential divergence

# 1.6 Network Architectures

- The structured neurons of a neural network are linked with the learning Algorithms (rules) used to train the network.
- Three are three fundamentally different classes of network architectures:
    1. Single-layer Feedforward Networks
    2. Multilayer Feedforward Networks
    3. Recurrent Networks

# 1. Single-layer Feedforward Networks

- In a layered Neural Network the neurons are organized in the form of layers.
  1. The input layer of source nodes
  2. The output layer of neurons (computation nodes)

This is illustrated in Fig. 1.15 for the case of 4 nodes both at input and output

- Such a network is called a single-layer network, referring to the output layer only
- The input layer of source nodes is not counted since no computation is performed at the input

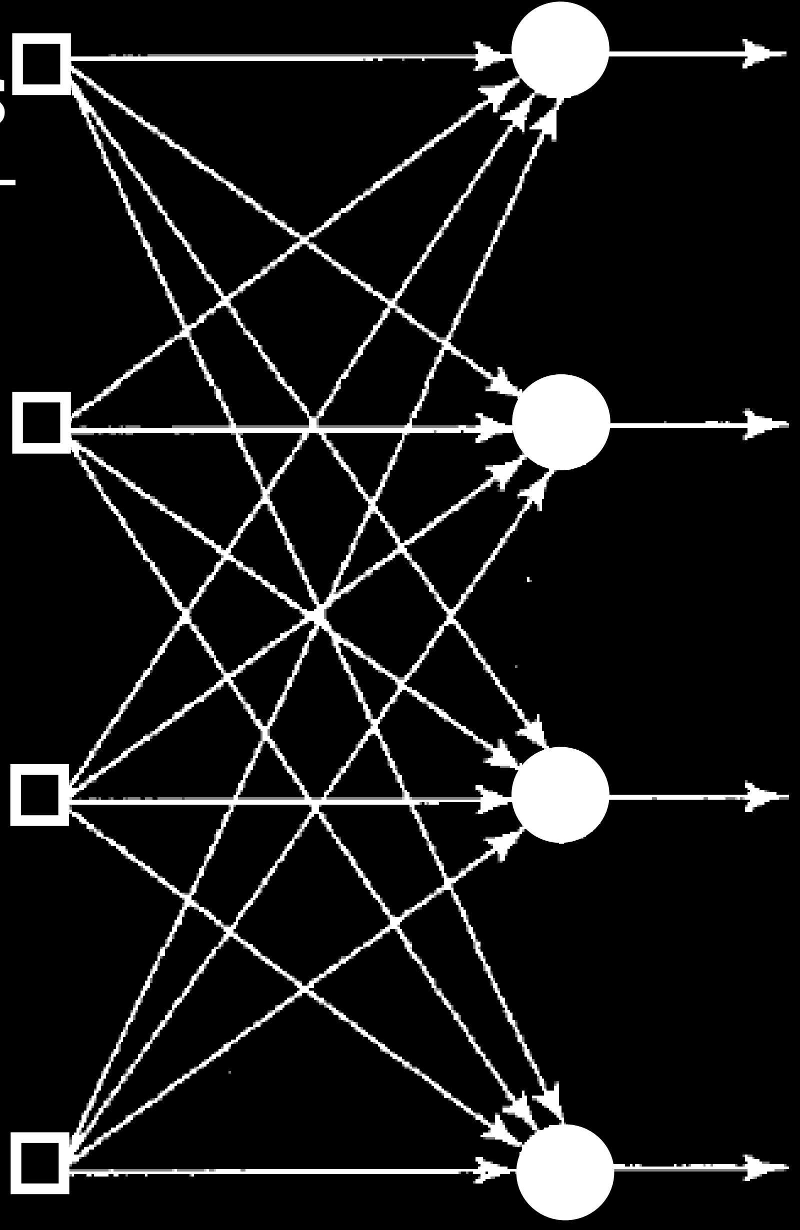


Fig. 1.15 Feedforward or acyclic network with a single layer of neurons.

Input layer of source nodes.

Output layer of neurons

# 2. Multilayer Feedforward Networks

- In Multilayer Feedforward Neural Networks, there are hidden layers.
- The computation nodes are called hidden neurons or hidden units
- The function of hidden neurons is to intervene between the external input and the network output
- By adding one or more hidden layers,
  - ~ the network is enabled to extract higher-order statistics
  - ~ due to the extra set of synaptic connections and the extra dimension of neural interactions

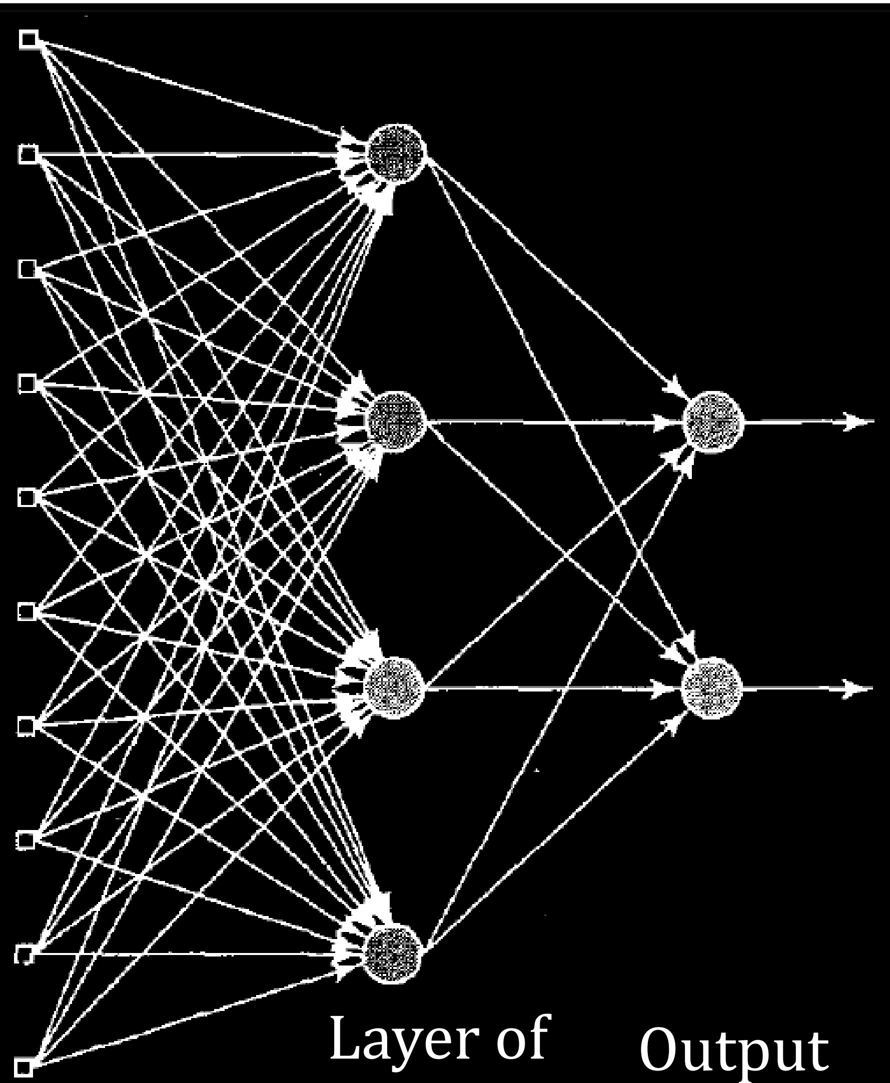
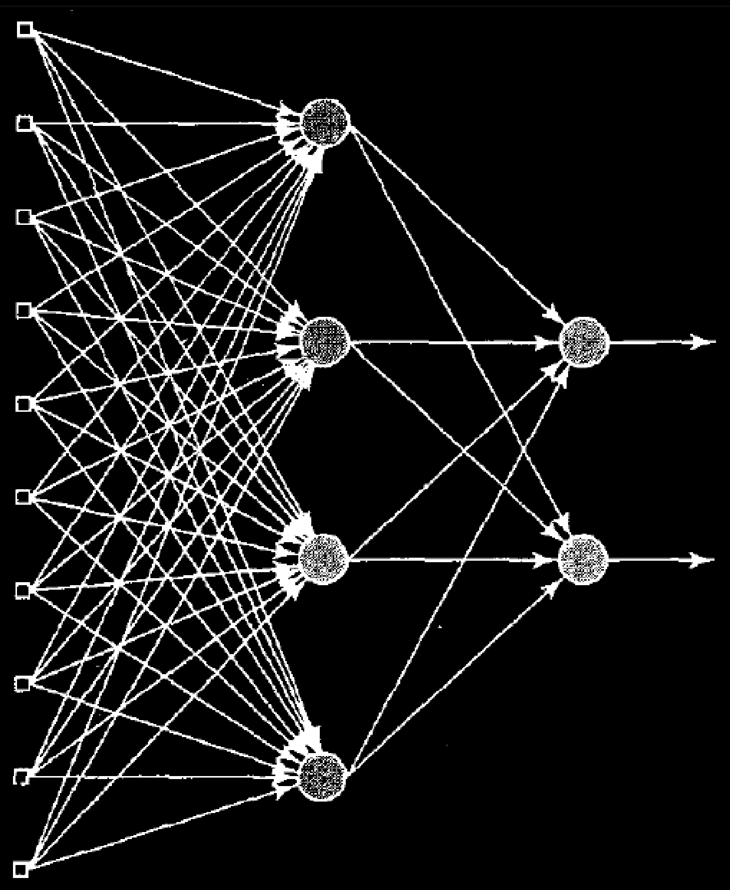



Fig. 1.16 Fully connected feedforward or acyclic network with one hidden layer and one output layer.

# 2. Multilayer Feedforward Networks

- The source nodes in the input layer of the network supply respective elements of the activation pattern (input vector), which constitute the input signals applied to the neurons (computation nodes) in the second layer
- The architectural graph in Fig. 1.16 illustrates the layout of a multilayer feedforward neural network for the case of a single hidden layer.
- The network in Fig. 1.16 is referred to as a 10-4-2 network because it has 10 source nodes, 4 hidden neurons and 2 output neurons
- Fig. 1.16 is said to be **fully connected network** every node in each layer of the network is connected c.f. **Partially connected networks**.



# 3. Recurrent Networks

- A recurrent neural network has at least one feedback loop
- For example, a recurrent network may consist of
  - ~ a single layer of neurons with each neuron feeding its output signal back to the inputs of all the other neurons
  - ~ there is no self-feedback loop in the network
  - ~ There is no hidden neurons
  - ~ as illustrated in the architectural the graph in Fig. 1.17

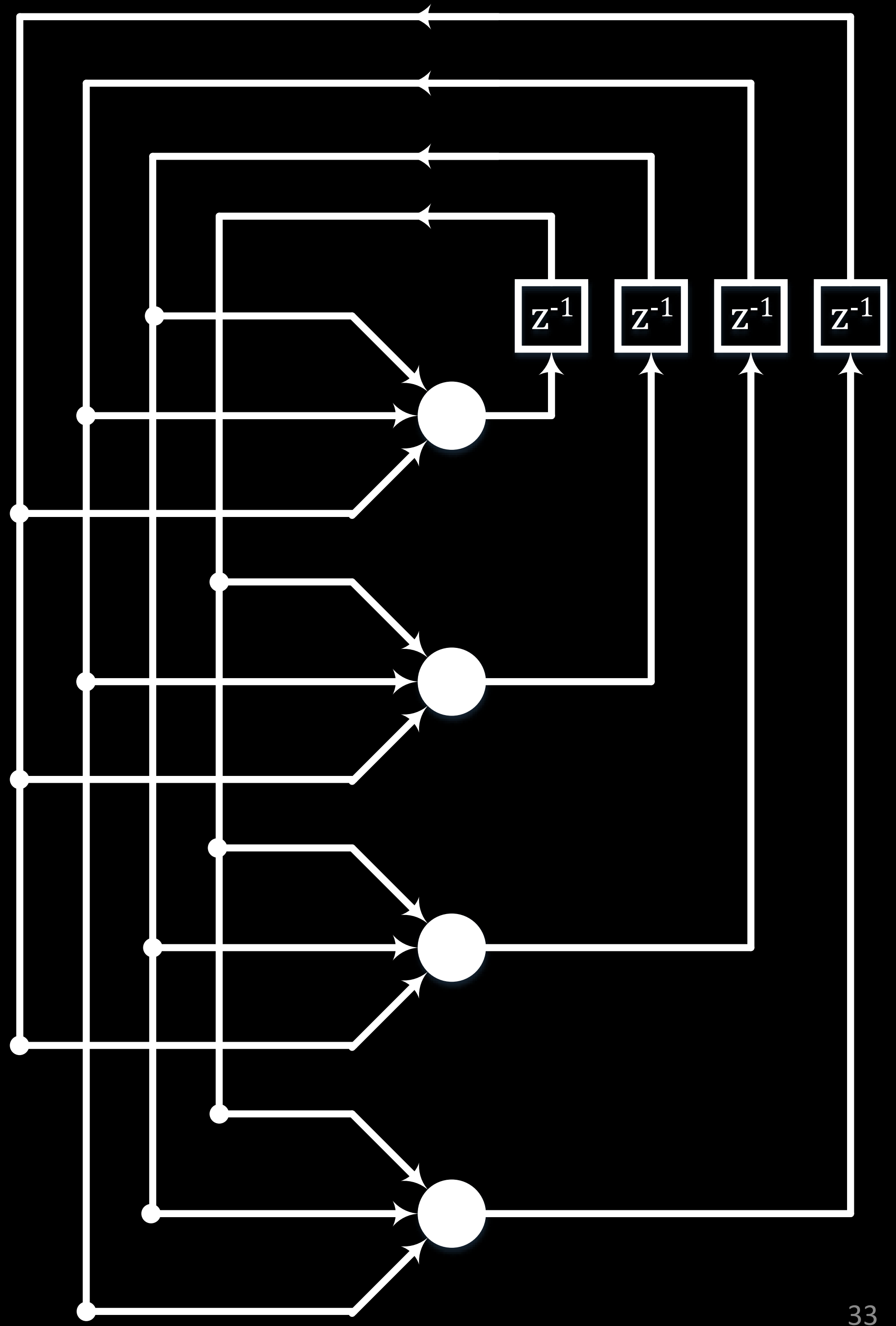



Fig. 1.17. Recurrent network with no self feedback loops and no hidden neurons

# 3. Recurrent Network with Hidden Neurons

- A recurrent network
  - ~ with hidden neurons and
  - ~ self-feedback
  - ~ where the connections originate from the hidden neurons as well as from the output neurons
  - ~ is shown in Fig. 1.18.
  - ~ The presence of feedback loops has a profound impact on the learning capability of the network and on its performance

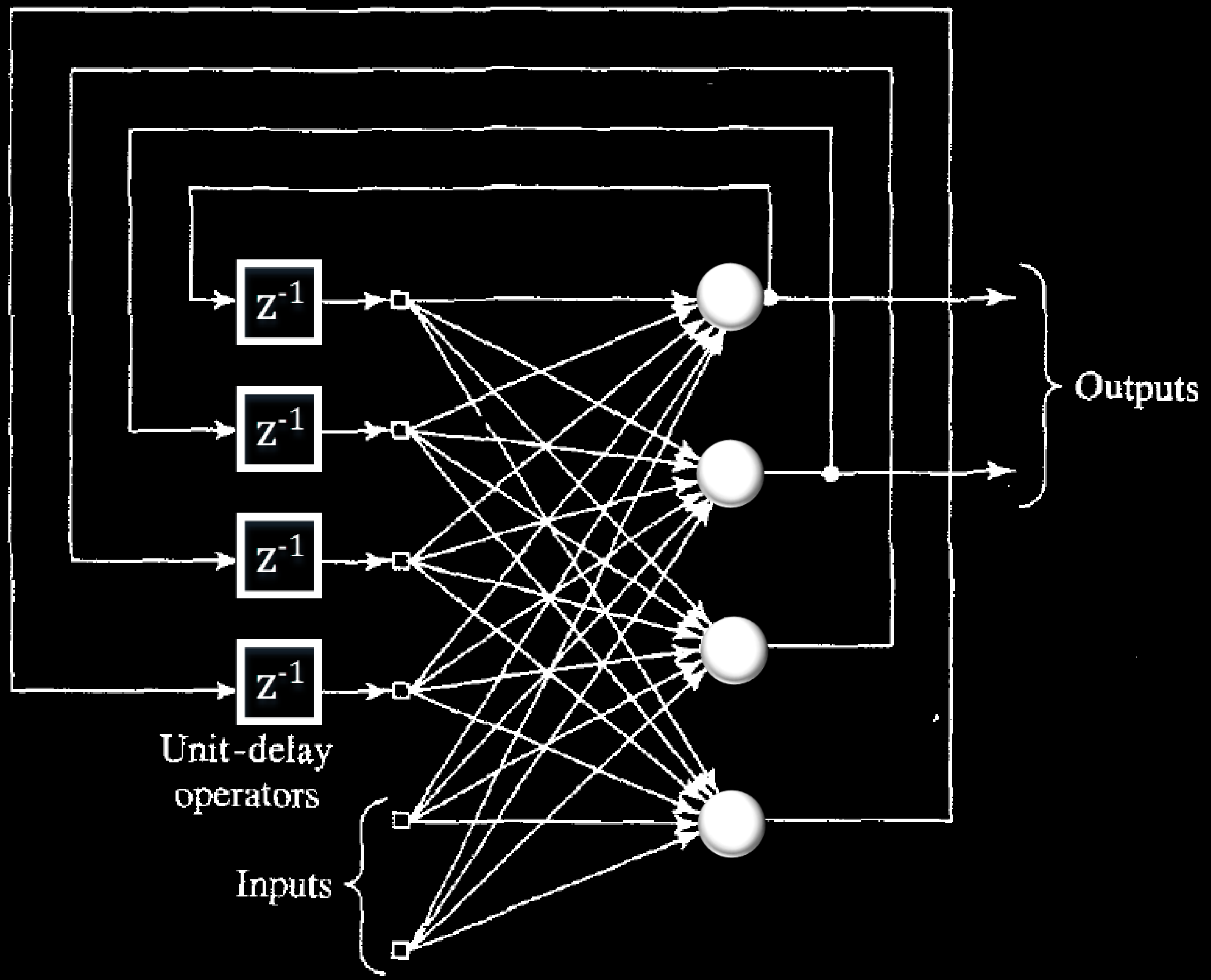



Fig. 1.18 Recurrent network with hidden neurons