CHAPTER 3

Elementary ANN Building Blocks

Everything should be made as simple as possible, but not simpler.

Albert Einstein

3.1 OVERVIEW AND OBJECTIVES

This chapter presents an extended look at the basic building block of ANNs. Individual neuron characteristics are developed, and the relationship of ANNs with biological neural systems is shown. In subsequent chapters we go from these individual neuron models to neural nets. Pedagogically speaking, understanding single units facilitates the understanding of the larger network.[1]

3.2 BIOLOGICAL NEURAL UNITS

In studying artificial neurons, it is helpful to first consider the biological origins of neurocomputing. In this section we explore the actual building blocks of biological neural systems. This area includes physiology, chemistry, and perception and can be viewed on many levels, from the subcell to the overall system. There are over 40 properties of biological neurons that influence their information-processing capability; therefore, only a summary explanation of this process is provided.

3.2.1 Physical (Biological) Neurons

Nerve cells

The nervous system consists of two classes of cells: *neurons*, or nerve cells, and *glia*, or glial cells. Neurons are the basic building blocks of biological information-processing systems. Glial cells perform more of a support function; therefore, we concentrate on neurons.

[1] At least it is a logical prerequisite.

The neurons of the brain may be classified according to function. *Afferent* or *sensory neurons* provide input to the nervous system; optic nerves are an example. *Motor neurons* transmit control signals to muscles and glands. *Interneuronal neurons* process information locally or propagate signals from one site to another, and constitute by far the largest class of cells in the nervous system.

A biological neuron typical of those found in vertebrates is shown in Figure 3.1. This cell has three major morphologically defined portions [Kan91], each of which makes a specific contribution to the processing of signals:

- The *cell body,* or *soma,* which consists of the cell nucleus and perikaryon. The cell body is typically 50 μm or larger in diameter.
- The *axon,* a tubular construct with a diameter ranging from 0.2 to 20 μm and with length up to 1 m. The axon begins at the *axon hillock,* which generates the cell action potential. The axon is the main conduction mechanism of the neuron.
- *Dendrites,* which branch out in treelike fashion. Most neurons have multiple dendrites. The dendrites of one neuron are connected to the axons of other neurons via *synaptic connections,* or synapses. This is how biological networks are formed. The multipolar neurons shown in Figure 3.1 have two types of dendrites, *apical* and *basal.* Basal dendrites facilitate both excitory and inhibitory functions in axon signal generation. We use this functionality in artificial cell models, especially the MP model (Section 3.3.4).

To facilitate discussion, we denote the cell that originates the signal as the *presynaptic* cell and the cell receiving the signal as the *postsynaptic cell*. This distinction is shown in Figure 3.1. The end of the axon divides into the main transmitting mechanisms of the neuron, the *presynaptic terminals*. The connection of the presynaptic neuron's axonic terminal to the dendrite of the postsynaptic neuron is called a *synapse*. There are usually between 1000 and 10,000 synapses on each neuron.

Cell morphology may be further classified according to the number of elements emanating from the cell body. On this basis, cells are subdivided into

- Unipolar cells, which have no dendrites emerging from the soma. A single primary process or branch exists and encompasses both dendrites and the axon. These cells are typical of the neurons found in invertebrates.
- Bipolar cells, which have two main processes. One contains dendrites (often in a sensory function), and the other consists of the axon.
- Multipolar cells, which are dominant in vertebrate nervous systems and have a single axon and one or more dendritic bundles. Multipolar cell structures are shown in Figure 3.2.

Synaptic activity

Synaptic transmission involves complicated chemical and electrical processes. Sensory or chemical stimuli initiate a change in synaptic potential. This is the basis by which one neuron influences the state of others. In the soma, this activity is integrated and determines axon potential. Note that both excitory and inhibitory influences are possible. The soma conversion from graded input action potentials to an all-or-nothing output is one of the most interesting aspects of cell behavior and gives rise to several artificial unit models. If the overall cell stimulus is below a threshold, no signal is

The neurons of the brain may be classified according to function. *Afferent* or *sensory neurons* provide input to the nervous system; optic nerves are an example. *Motor neurons* transmit control signals to muscles and glands. *Interneuronal neurons* process information locally or propagate signals from one site to another, and constitute by far the largest class of cells in the nervous system.

A biological neuron typical of those found in vertebrates is shown in Figure 3.1. This cell has three major morphologically defined portions [Kan91], each of which makes a specific contribution to the processing of signals:

- The *cell body,* or *soma,* which consists of the cell nucleus and perikaryon. The cell body is typically 50 μm or larger in diameter.
- The *axon,* a tubular construct with a diameter ranging from 0.2 to 20 μm and with length up to 1 m. The axon begins at the *axon hillock,* which generates the cell action potential. The axon is the main conduction mechanism of the neuron.
- *Dendrites,* which branch out in treelike fashion. Most neurons have multiple dendrites. The dendrites of one neuron are connected to the axons of other neurons via *synaptic connections,* or synapses. This is how biological networks are formed. The multipolar neurons shown in Figure 3.1 have two types of dendrites, *apical* and *basal.* Basal dendrites facilitate both excitory and inhibitory functions in axon signal generation. We use this functionality in artificial cell models, especially the MP model (Section 3.3.4).

To facilitate discussion, we denote the cell that originates the signal as the *presynaptic* cell and the cell receiving the signal as the *postsynaptic cell.* This distinction is shown in Figure 3.1. The end of the axon divides into the main transmitting mechanisms of the neuron, the *presynaptic terminals*. The connection of the presynaptic neuron's axonic terminal to the dendrite of the postsynaptic neuron is called a *synapse*. There are usually between 1000 and 10,000 synapses on each neuron.

Cell morphology may be further classified according to the number of elements emanating from the cell body. On this basis, cells are subdivided into

- Unipolar cells, which have no dendrites emerging from the soma. A single primary process or branch exists and encompasses both dendrites and the axon. These cells are typical of the neurons found in invertebrates.
- Bipolar cells, which have two main processes. One contains dendrites (often in a sensory function), and the other consists of the axon.
- Multipolar cells, which are dominant in vertebrate nervous systems and have a single axon and one or more dendritic bundles. Multipolar cell structures are shown in Figure 3.2.

Synaptic activity

Synaptic transmission involves complicated chemical and electrical processes. Sensory or chemical stimuli initiate a change in synaptic potential. This is the basis by which one neuron influences the state of others. In the soma, this activity is integrated and determines axon potential. Note that both excitory and inhibitory influences are possible. The soma conversion from graded input action potentials to an all-or-nothing output is one of the most interesting aspects of cell behavior and gives rise to several artificial unit models. If the overall cell stimulus is below a threshold, no signal is

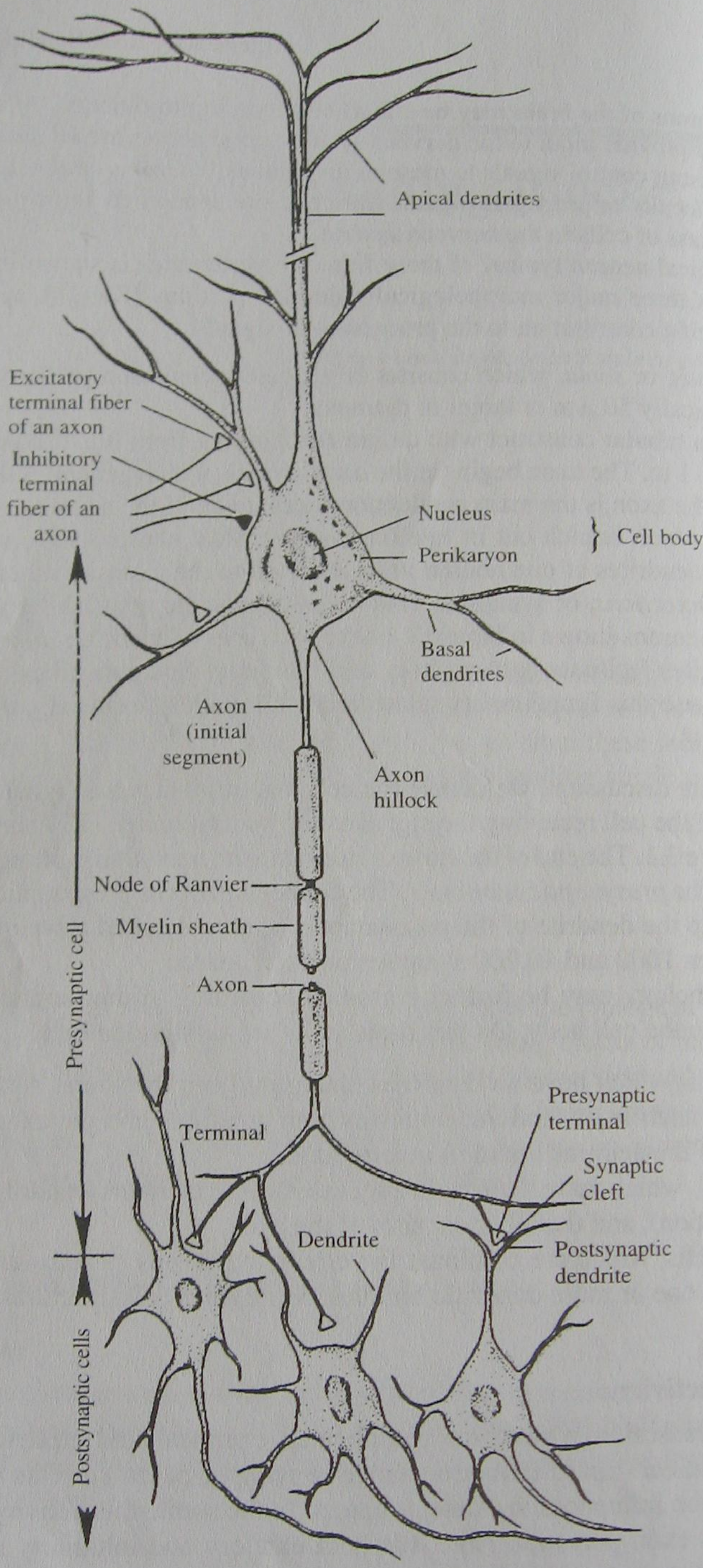


FIGURE 3.1
Expanded view of single neuron morphology [Kan91].

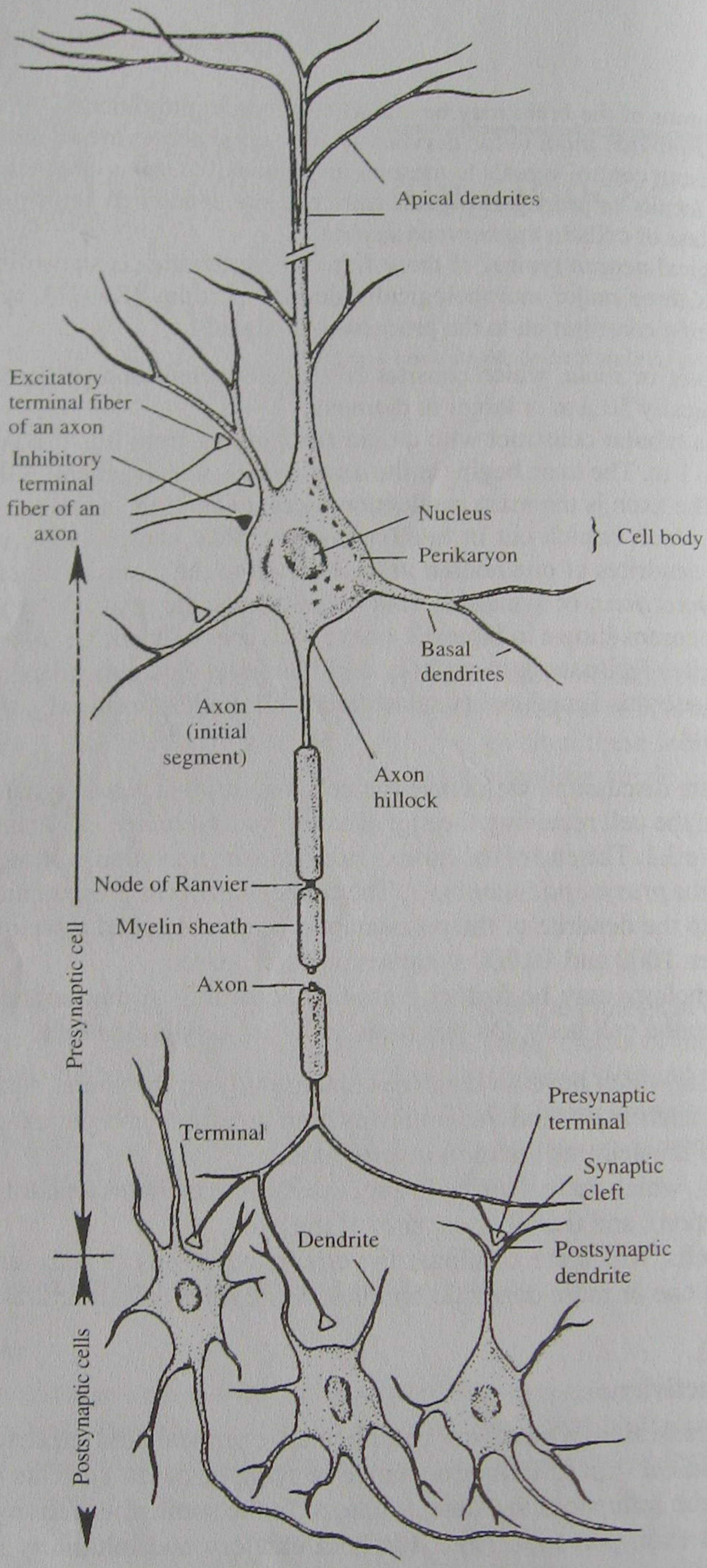


FIGURE 3.1
Expanded view of single neuron morphology [Kan91].

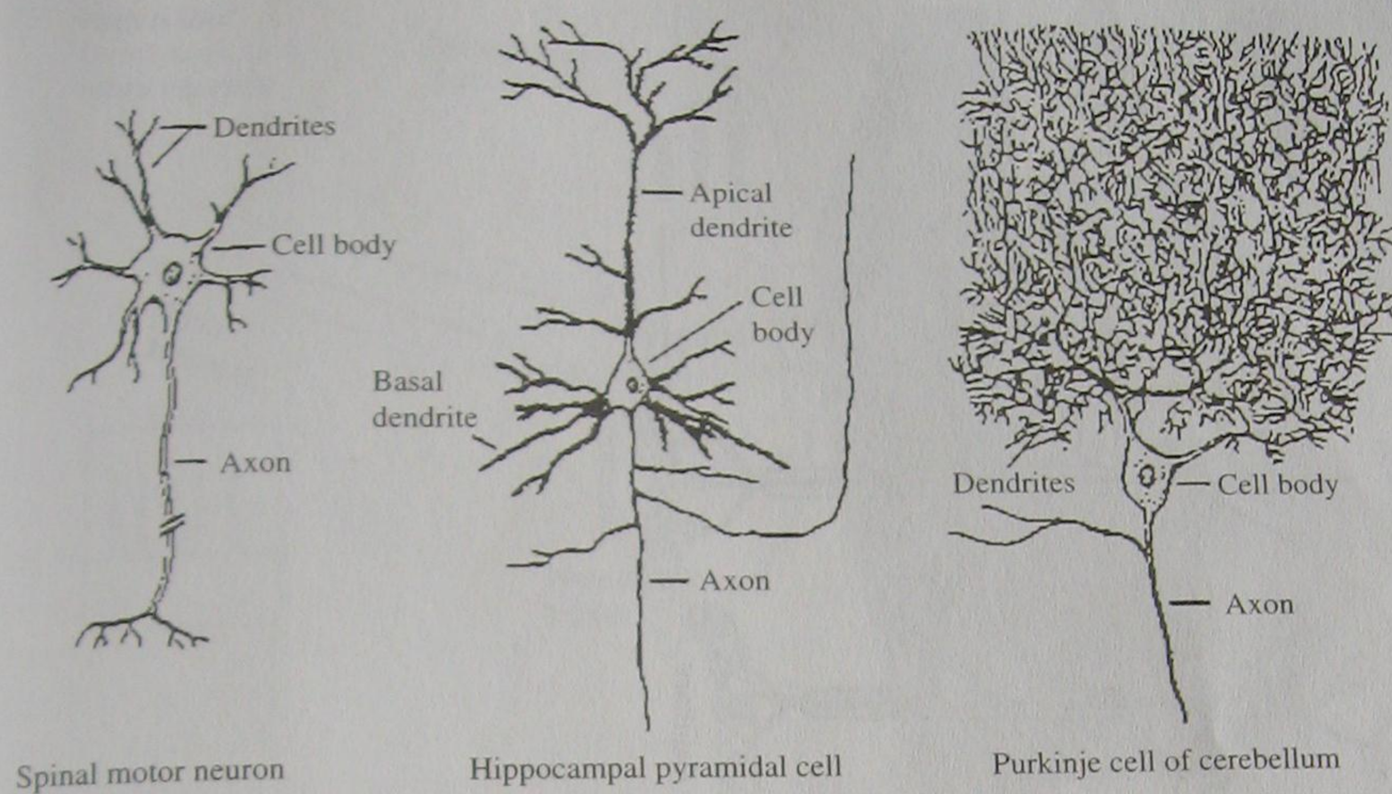


FIGURE 3.2
Examples of multipolar cells [Kan91].

produced. If the accumulated stimulus is above the threshold, *regardless of how much above*, the same output is produced. This process is generic and relatively independent of cell type. Examples using several different neuron types are shown in Figure 3.3.

The *action potential* for an activated neuron is usually a spiked signal where the frequency is proportional to the potential of the soma. If the neuron's soma potential rises above some threshold value, the neuron begins firing. An action potential, therefore, may cause changes in the potential of attached neurons. The average frequency of the action potential is known as the *mean firing rate* of the neuron. The mean soma potential with respect to the mean resting soma potential is known as the *activation level* of the neuron. This is shown in Figure 3.4. Table 3.1, excerpted from [Kan91], illustrates specific parameters of the electroneural process.

Some neurotransmitters are excitatory, which means that they cause an increase in the soma potential of the receiving neuron, and some are inhibitory, which means that they either lower the receiving neuron's soma potential or prevent it from increasing. A special case is presynaptic inhibition, caused by a synapse appearing on the presynaptic nerve fiber or the synaptic knob. This form of inhibition appears to result in a substantial reduction of the action potential magnitude at the synapse. The net result is a multiplicative effect on the transfer of activation. Postsynaptic inhibition is a negative feedback mechanism used in preventing the excessive spread of activation.

3.2.2 The Scale of Biological Systems

In Chapter 1 we introduced the problems associated with scaling of ANN solutions. It is something of an understatement to say that nature has solved the "scaling problem" as far as neural networks are concerned. Table 3.2, from [CS93], gives some idea of

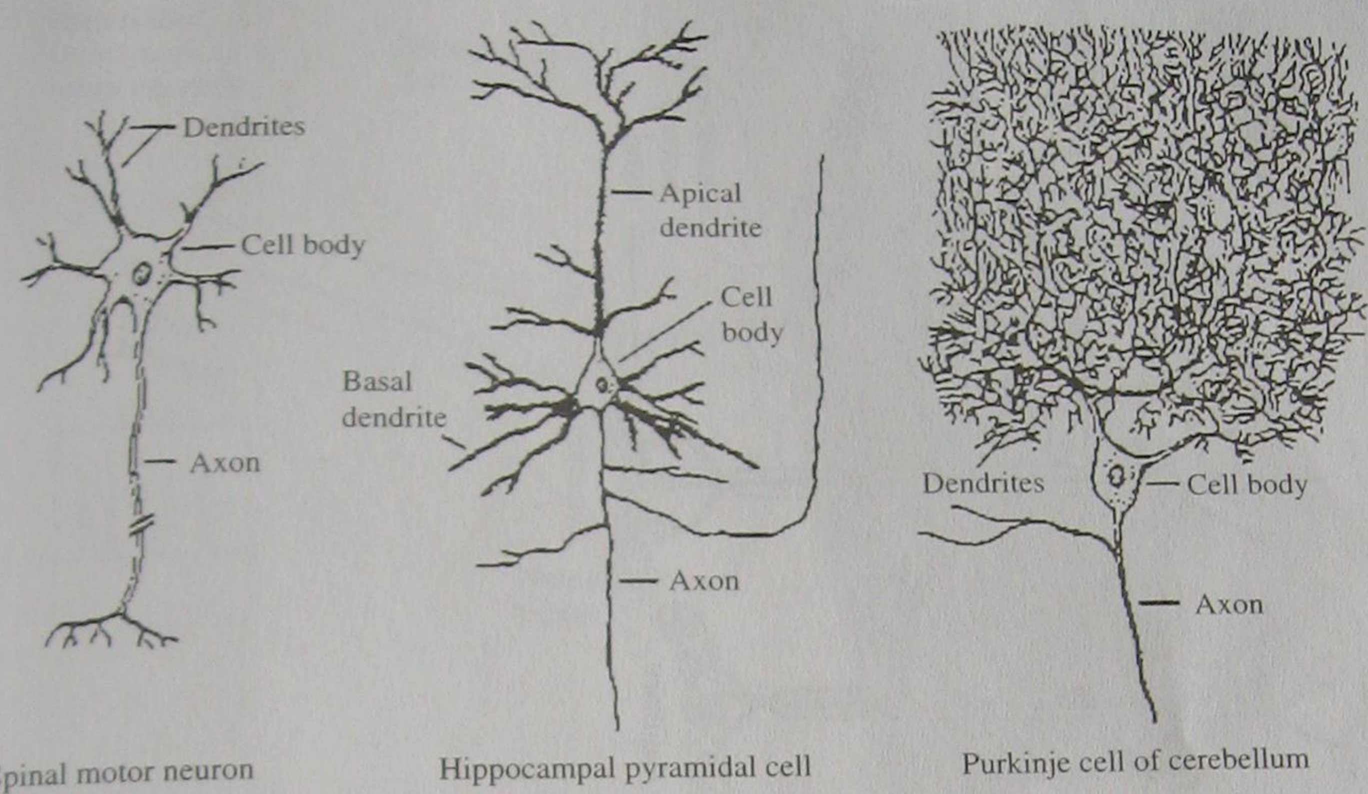


FIGURE 3.2
Examples of multipolar cells [Kan91].

produced. If the accumulated stimulus is above the threshold, *regardless of how much above*, the same output is produced. This process is generic and relatively independent of cell type. Examples using several different neuron types are shown in Figure 3.3.

The *action potential* for an activated neuron is usually a spiked signal where the frequency is proportional to the potential of the soma. If the neuron's soma potential rises above some threshold value, the neuron begins firing. An action potential, therefore, may cause changes in the potential of attached neurons. The average frequency of the action potential is known as the *mean firing rate* of the neuron. The mean soma potential with respect to the mean resting soma potential is known as the *activation level* of the neuron. This is shown in Figure 3.4. Table 3.1, excerpted from [Kan91], illustrates specific parameters of the electroneural process.

Some neurotransmitters are excitatory, which means that they cause an increase in the soma potential of the receiving neuron, and some are inhibitory, which means that they either lower the receiving neuron's soma potential or prevent it from increasing. A special case is presynaptic inhibition, caused by a synapse appearing on the presynaptic nerve fiber or the synaptic knob. This form of inhibition appears to result in a substantial reduction of the action potential magnitude at the synapse. The net result is a multiplicative effect on the transfer of activation. Postsynaptic inhibition is a negative feedback mechanism used in preventing the excessive spread of activation.

3.2.2 The Scale of Biological Systems

In Chapter 1 we introduced the problems associated with scaling of ANN solutions. It is something of an understatement to say that nature has solved the "scaling problem" as far as neural networks are concerned. Table 3.2, from [CS93], gives some idea of

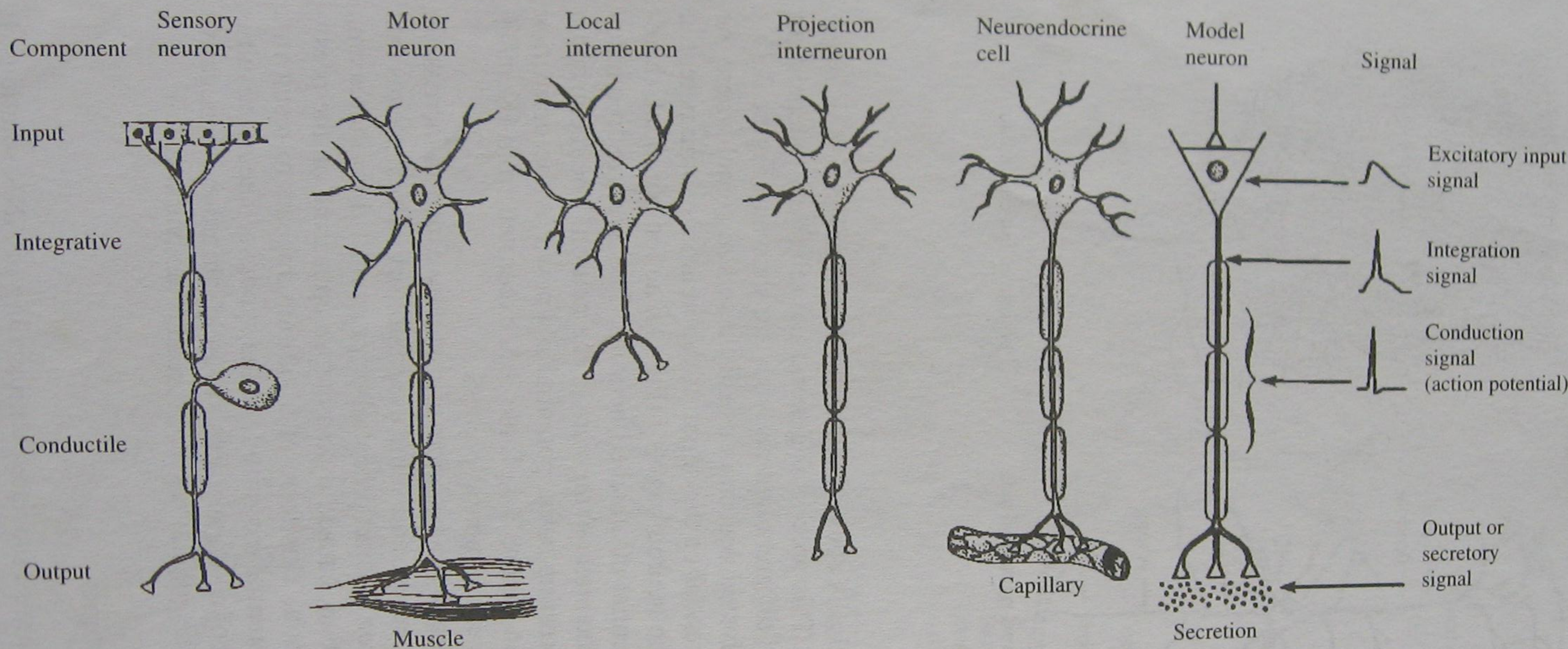


FIGURE 3.3
Four functional cell components [Kan91].

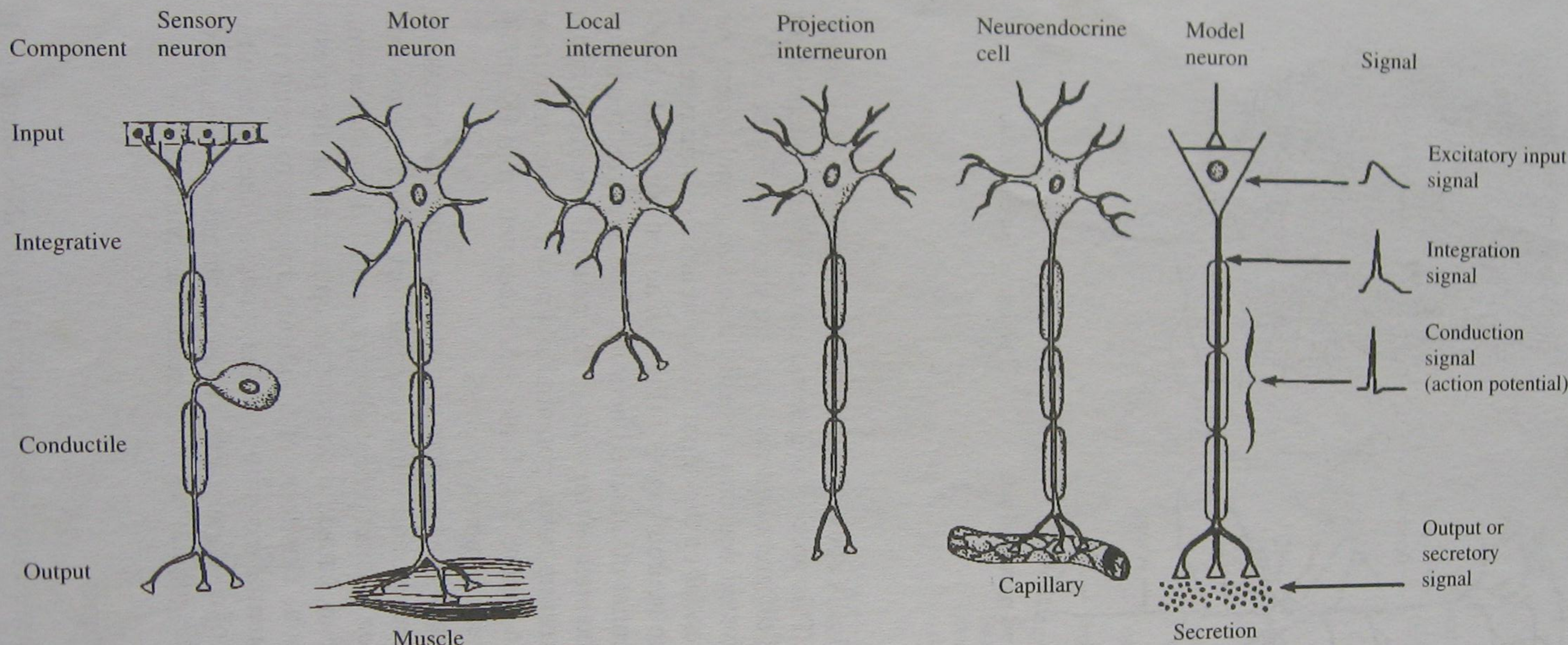


FIGURE 3.3
Four functional cell components [Kan91].

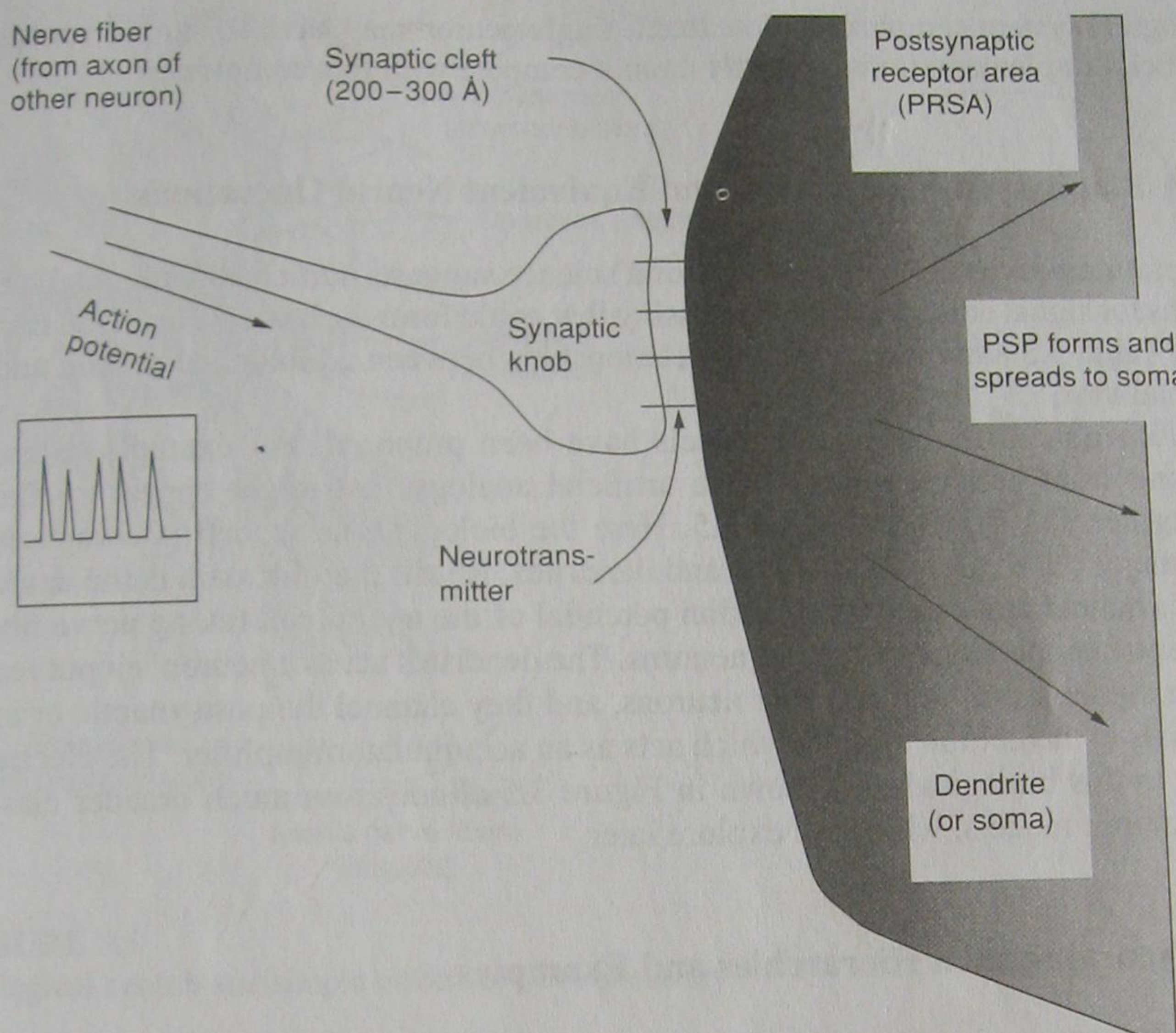


FIGURE 3.4
Activation and synaptic firing in a biological neuron.

TABLE 3.1
Receptor, synaptic, and action potentials

Feature	Receptor potential	Synaptic potential	Action potential
Amplitude	0.1–10 mV	0.1–10 mV	70–110 mV
Duration	5–100 ms	5 ms to 20 min	1–10 ms
Resolution	Graded (continuous)	Graded	Bilevel (all or none)

TABLE 3.2
Approximate numbers of neurons and synapses in two nervous systems

System	Neurons	Synapses
Human nervous system	10^{12}	10^{15}
Rat brain	10^{10}	10^{13}

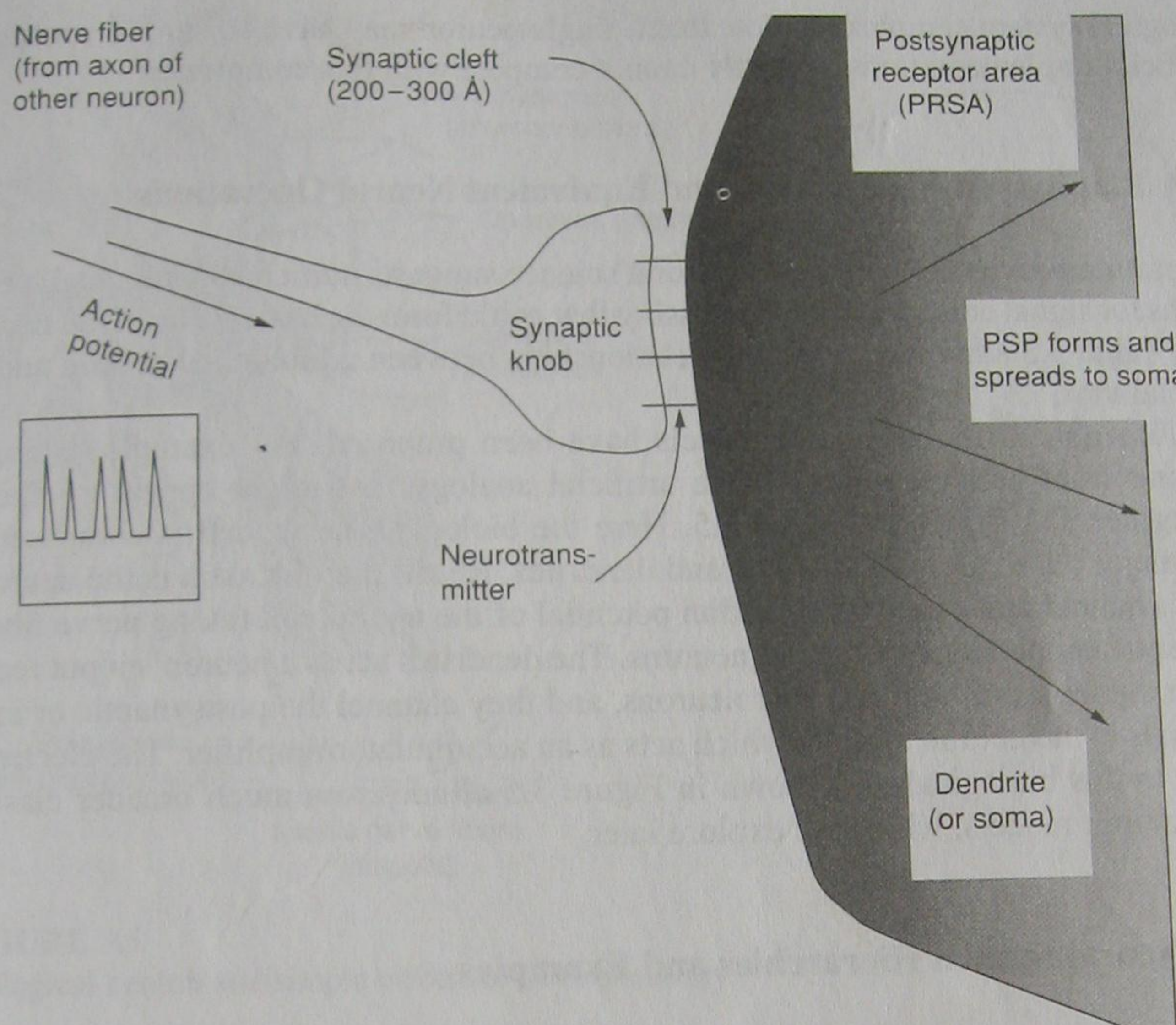


FIGURE 3.4
Activation and synaptic firing in a biological neuron.

TABLE 3.1
Receptor, synaptic, and action potentials

Feature	Receptor potential	Synaptic potential	Action potential
Amplitude	0.1–10 mV	0.1–10 mV	70–110 mV
Duration	5–100 ms	5 ms to 20 min	1–10 ms
Resolution	Graded (continuous)	Graded	Bilevel (all or none)

TABLE 3.2
Approximate numbers of neurons and synapses in two nervous systems

System	Neurons	Synapses
Human nervous system	10^{12}	10^{15}
Rat brain	10^{10}	10^{13}

biological system complexity. Note that a single neuron may have 10^3 to 10^4 synapses. Artificial implementations currently cannot compete with this complexity.

3.2.3 Biophysical Mechanisms and Equivalent Neural Operations

Our previous review of elements of neural science suggests numerous biological mechanisms for signal or information processing that could form the basis for artificial neuron units. Table 3.3 summarizes typical relationships between a biological neuron and an artificial unit.

Numerous artificial neuron models have been proposed. For example, a simple mapping from biological units to an artificial analogy that might appeal to electrical engineers is shown in Figure 3.5. Here the biological nerve cell is composed of three major parts: a soma, an axon, and dendrites. Recall that the axon is the neuron's output channel and conveys the action potential of the neural cell (along nerve fibers) to synaptic connections with other neurons. The dendrites act as a neuron's input receptors for signals coming from other neurons, and they channel the postsynaptic or input potentials to the neuron's soma, which acts as an accumulator/amplifier. The electronic analog to this biological unit shown in Figure 3.5 alludes to a much broader class of artificial unit models, which we explore later.

3.2.4 Neural System Hierarchies and Examples

Efforts to unify the microscopic activity of neural systems with cognitive skills abound [Chu86]. Although intelligent behavior of biological systems is viewed on a macroscopic scale, underlying this behavior is the fusion of thousands to billions of microscopic biological processing units, i.e., neurons.

The following are broad clues about the computation in high-level nervous systems and especially about the human brain:

- The brain is not like a general-purpose computer; rather, it is a special-purpose machine that is efficient at some tasks but limited in flexibility.
- The brain is the product of evolution, not engineering. Changes are *incremental*; i.e., the brain cannot start over "from scratch" as in engineering design.

A proposed organization of higher-level biological systems is shown in Figure 3.6.

TABLE 3.3
Comparison of physical and artificial neurons

Physical (biological)	Artificial
(Neuron) cell	Unit
Synapse	Interconnection weight
Excitatory input	(Large) positive interconnection weight
Inhibitory input	(Large) negative interconnection weight
Activation by (spiking) frequency	DC level
Range of activation limited by cell physics	Range of activation limited by squashing function

biological system complexity. Note that a single neuron may have 10^3 to 10^4 synapses. Artificial implementations currently cannot compete with this complexity.

3.2.3 Biophysical Mechanisms and Equivalent Neural Operations

Our previous review of elements of neural science suggests numerous biological mechanisms for signal or information processing that could form the basis for artificial neuron units. Table 3.3 summarizes typical relationships between a biological neuron and an artificial unit.

Numerous artificial neuron models have been proposed. For example, a simple mapping from biological units to an artificial analogy that might appeal to electrical engineers is shown in Figure 3.5. Here the biological nerve cell is composed of three major parts: a soma, an axon, and dendrites. Recall that the axon is the neuron's output channel and conveys the action potential of the neural cell (along nerve fibers) to synaptic connections with other neurons. The dendrites act as a neuron's input receptors for signals coming from other neurons, and they channel the postsynaptic or input potentials to the neuron's soma, which acts as an accumulator/amplifier. The electronic analog to this biological unit shown in Figure 3.5 alludes to a much broader class of artificial unit models, which we explore later.

3.2.4 Neural System Hierarchies and Examples

Efforts to unify the microscopic activity of neural systems with cognitive skills abound [Chu86]. Although intelligent behavior of biological systems is viewed on a macroscopic scale, underlying this behavior is the fusion of thousands to billions of microscopic biological processing units, i.e., neurons.

The following are broad clues about the computation in high-level nervous systems and especially about the human brain:

- The brain is not like a general-purpose computer; rather, it is a special-purpose machine that is efficient at some tasks but limited in flexibility.
- The brain is the product of evolution, not engineering. Changes are *incremental*; i.e., the brain cannot start over "from scratch" as in engineering design.

A proposed organization of higher-level biological systems is shown in Figure 3.6.

TABLE 3.3
Comparison of physical and artificial neurons

Physical (biological)	Artificial
(Neuron) cell	Unit
Synapse	Interconnection weight
Excitatory input	(Large) positive interconnection weight
Inhibitory input	(Large) negative interconnection weight
Activation by (spiking) frequency	DC level
Range of activation limited by cell physics	Range of activation limited by squashing function

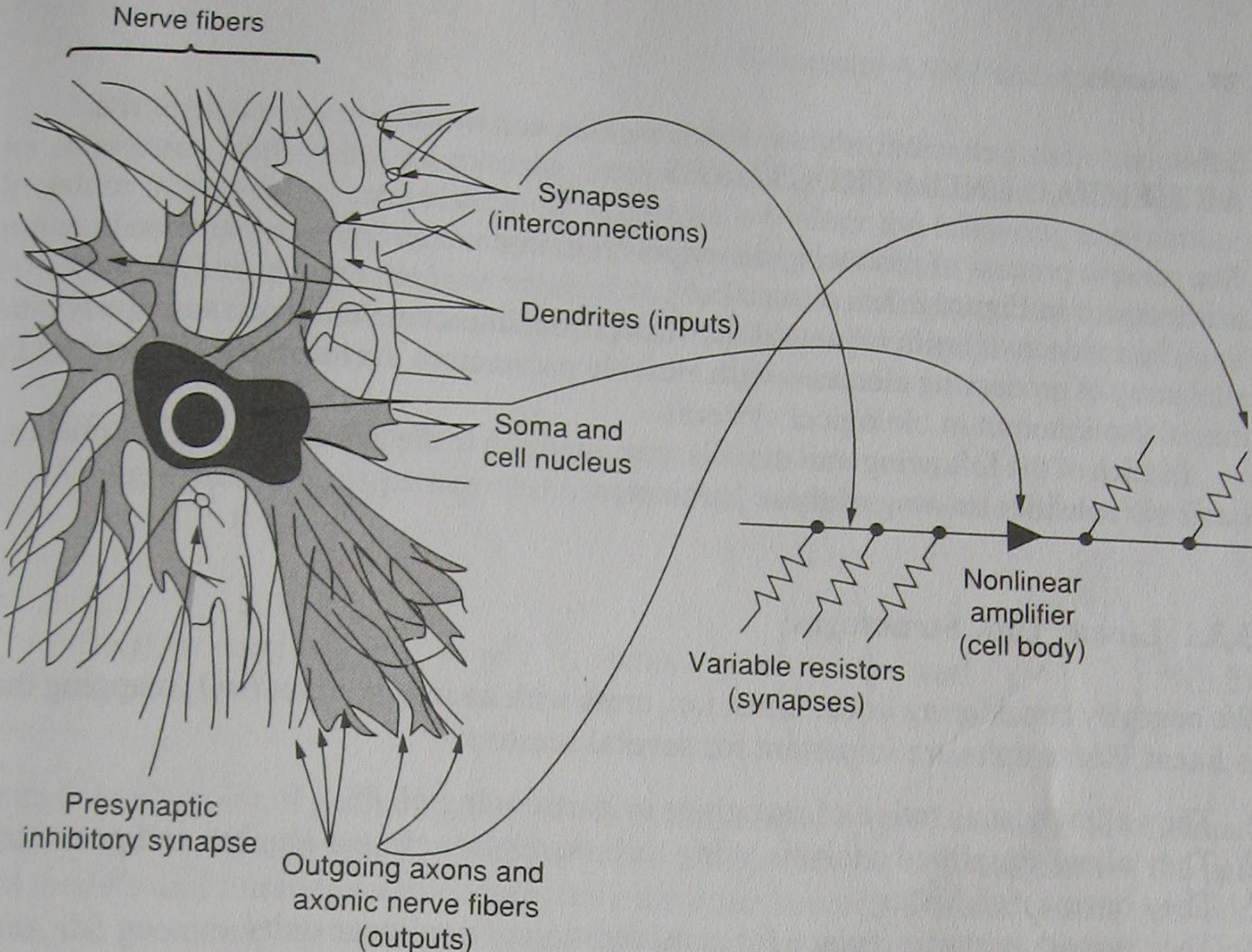


FIGURE 3.5
Biological neuron and simple electrical device analogy.

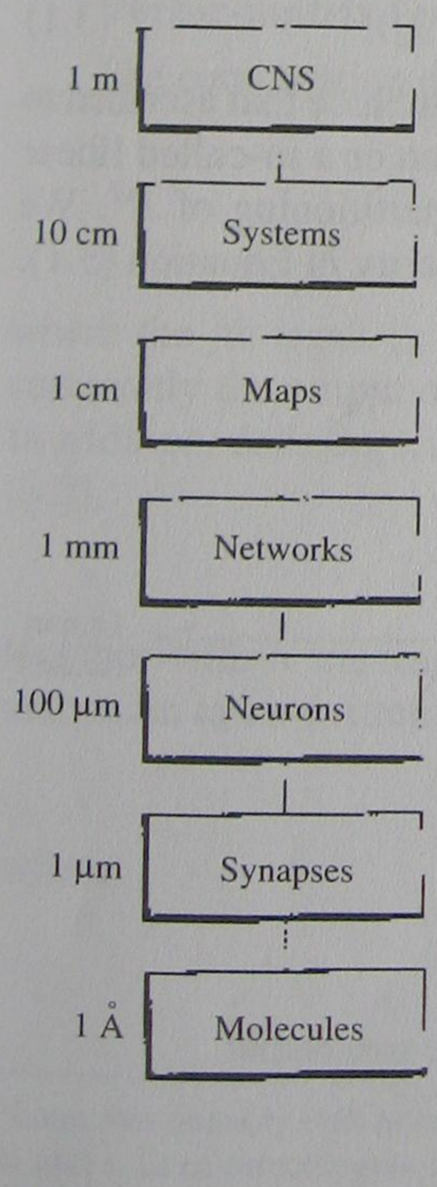


FIGURE 3.6
Levels of organization in the nervous system [CS93].

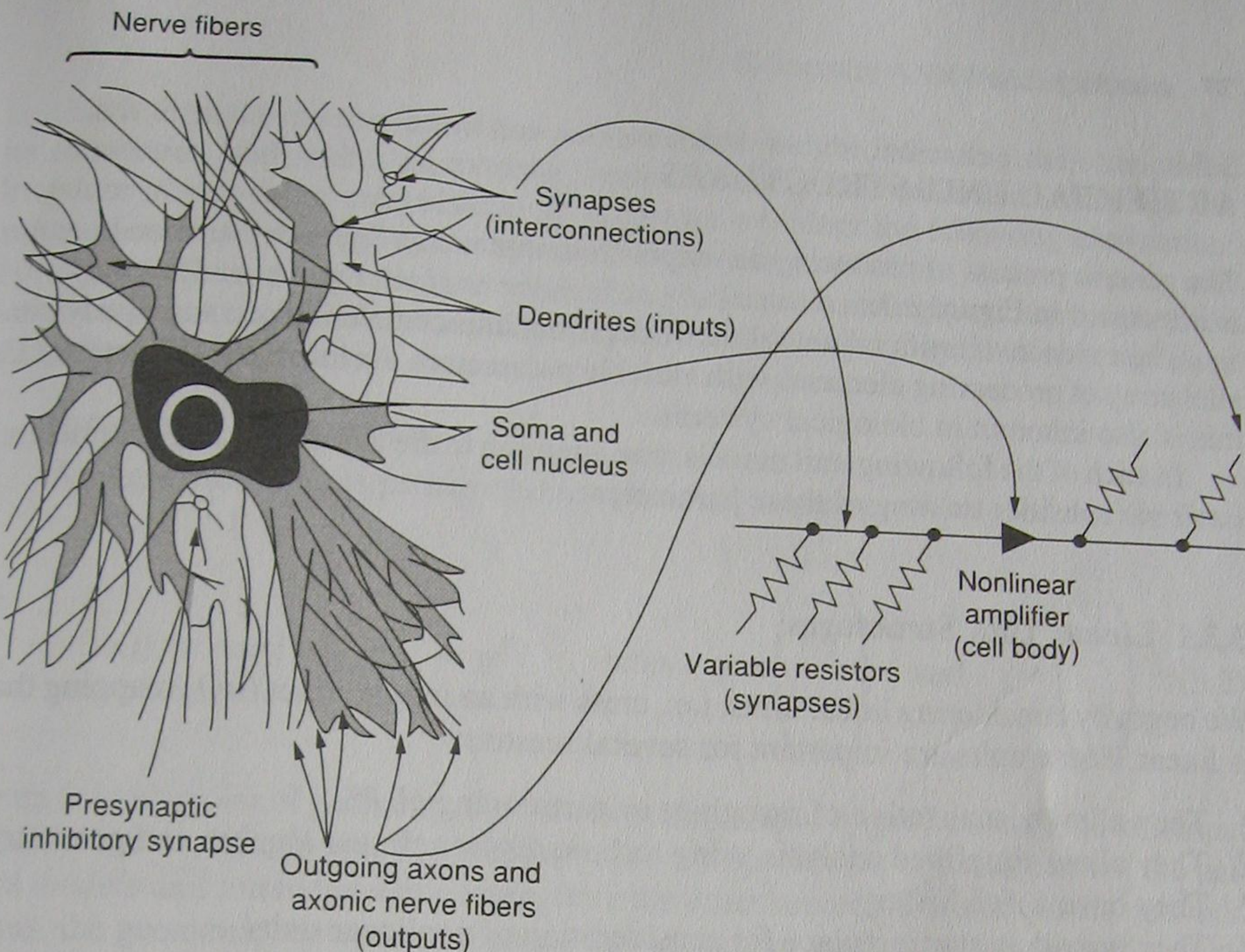


FIGURE 3.5
Biological neuron and simple electrical device analogy.

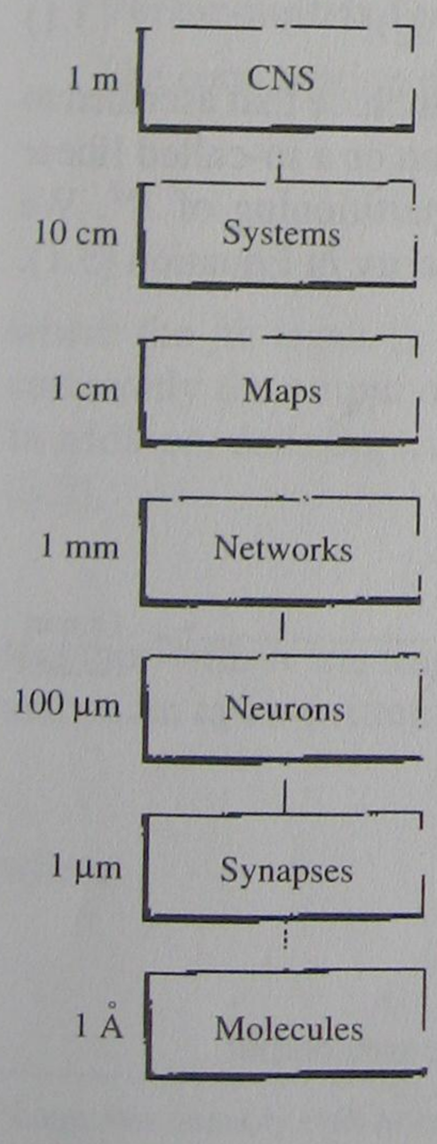


FIGURE 3.6
Levels of organization in the nervous system [CS93].

3.3 ARTIFICIAL UNIT STRUCTURES

The generic process of producing an output from inputs to a "generic" artificial neuron is illustrated in Figure 3.7.

The essence of artificial neural networks is the interconnection of a massively parallel array of processing elements with variable parameters. As indicated in Section 3.2, this is also inherent in biological systems.

In each of the following unit models, pay attention to the adjustable unit parameters. Later we consider training of these parameters.

3.3.1 Linear Unit Structures

We begin by considering linear units, i.e., units with an input/output (I/O) mapping that is linear. These units are important for several reasons:

1. They allow visualization of mappings or partitioning of R^d.
2. They allow simplified analysis using techniques from linear algebra and geometry.
3. They have a rich history.
4. They provide a starting place for consideration of nonlinear units.
5. In themselves, they provide many useful mappings (Chapters 4 and 5).

Recall from Chapter 2 that a mapping, $\underline{f}(\underline{x})$, is linear in the input/output sense if

$$\underline{x} = \alpha \underline{x}_1 + \beta \underline{x}_2 \Rightarrow \underline{f}(\underline{x}) = \alpha \underline{f}(\underline{x}_1) + \beta \underline{f}(\underline{x}_2) \tag{3.1}$$

Other characterizations of linear behavior found in the literature include that ascribed to a unit whose I/O characteristic implements a hyperplane equation or a so-called linear decision boundary. These units yield a "linearly separable"[2] partitioning of R^d. We should not, however, lose sight of the strict definition of I/O linearity in Equation (3.1).

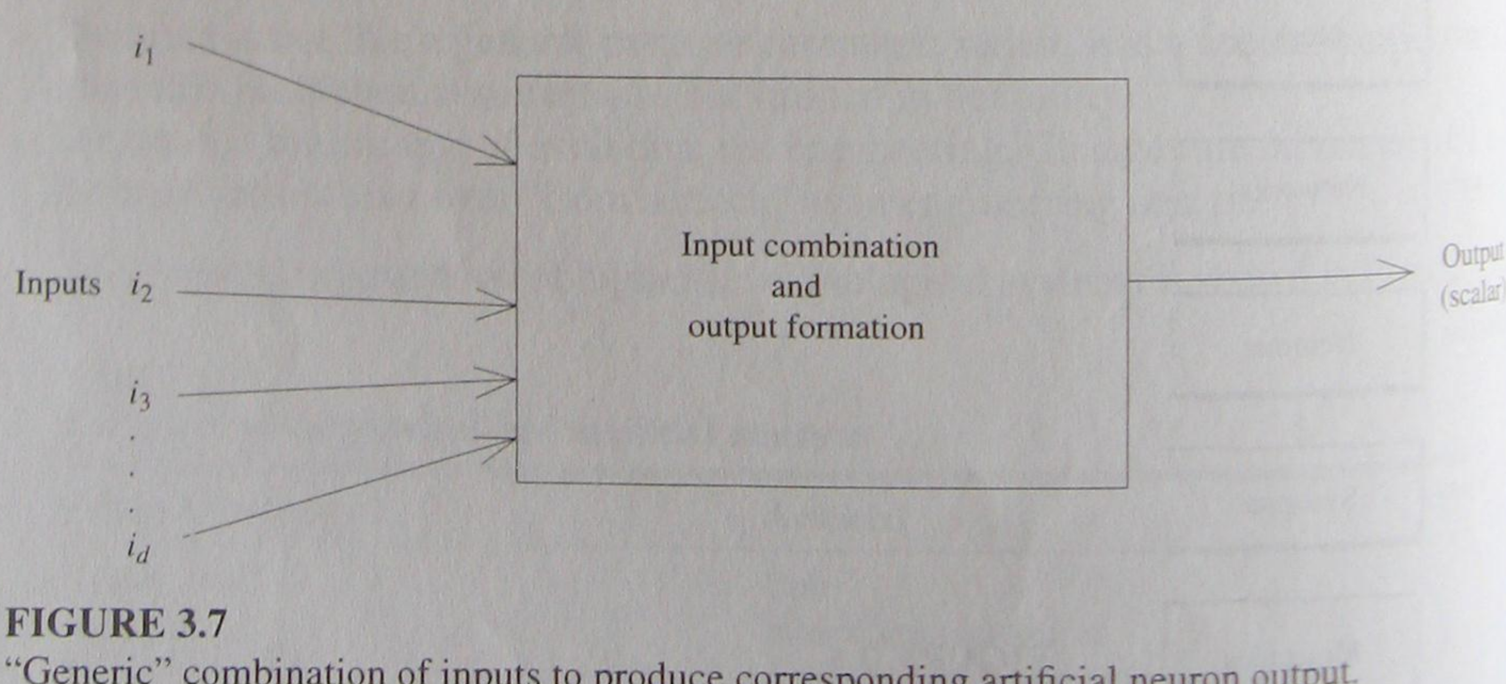


FIGURE 3.7
"Generic" combination of inputs to produce corresponding artificial neuron output.

[2]Often using nonlinear units.

3.3 ARTIFICIAL UNIT STRUCTURES

The generic process of producing an output from inputs to a "generic" artificial neuron is illustrated in Figure 3.7.

The essence of artificial neural networks is the interconnection of a massively parallel array of processing elements with variable parameters. As indicated in Section 3.2, this is also inherent in biological systems.

In each of the following unit models, pay attention to the adjustable unit parameters. Later we consider training of these parameters.

3.3.1 Linear Unit Structures

We begin by considering linear units, i.e., units with an input/output (I/O) mapping that is linear. These units are important for several reasons:

1. They allow visualization of mappings or partitioning of R^d.
2. They allow simplified analysis using techniques from linear algebra and geometry.
3. They have a rich history.
4. They provide a starting place for consideration of nonlinear units.
5. In themselves, they provide many useful mappings (Chapters 4 and 5).

Recall from Chapter 2 that a mapping, $\underline{f}(\underline{x})$, is linear in the input/output sense if

$$\underline{x} = \alpha \underline{x}_1 + \beta \underline{x}_2 \Rightarrow \underline{f}(\underline{x}) = \alpha f(\underline{x}_1) + \beta f(\underline{x}_2) \tag{3.1}$$

Other characterizations of linear behavior found in the literature include that ascribed to a unit whose I/O characteristic implements a hyperplane equation or a so-called linear decision boundary. These units yield a "linearly separable"[2] partitioning of R^d. We should not, however, lose sight of the strict definition of I/O linearity in Equation (3.1).

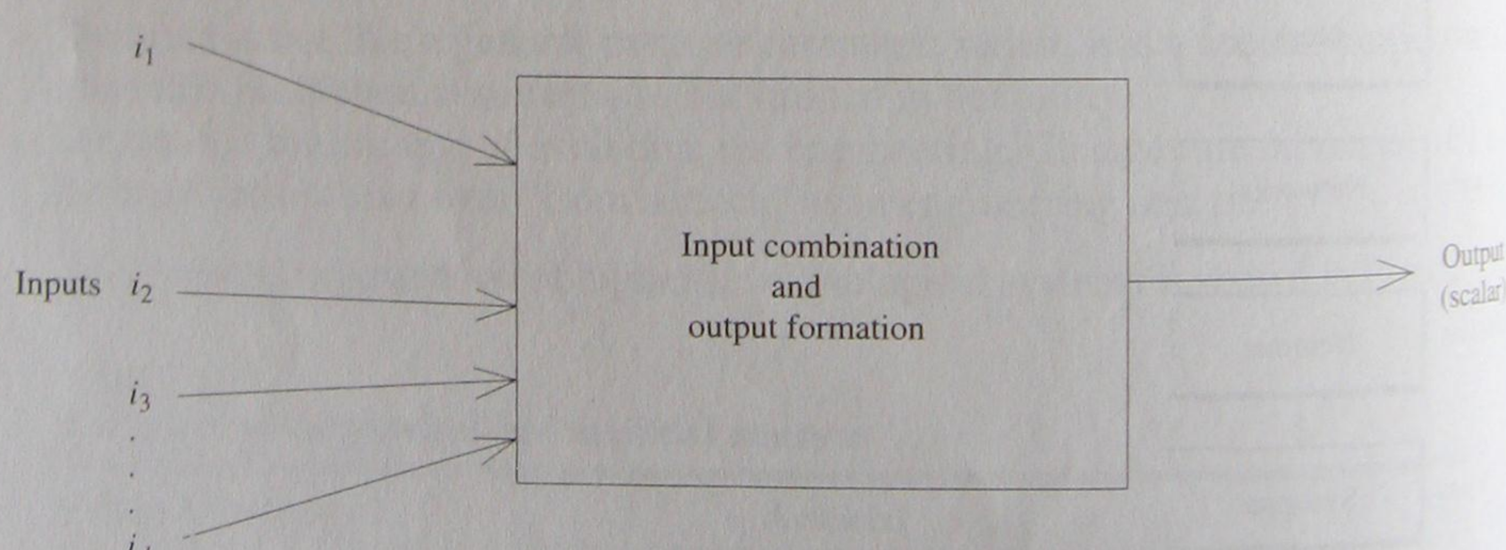


FIGURE 3.7
"Generic" combination of inputs to produce corresponding artificial neuron output.

[2]Often using nonlinear units.

Each artificial neuron input has an associated weight indicating the strength of its connection with either an external input or another neuron output. Although the literature is somewhat inconsistent on this topic, we adopt the following convention: *w_{ij} represents the strength of the connection* to *neuron unit i* from *(either) neuron unit j or input j*. Thus, a large positive value of w_{ij} indicates a strongly excitory input from unit or input j to unit i, whereas a large negative weight value may be used to represent a highly inhibitory input.

Single-unit formulation

A single linear unit is perhaps the simplest S-R mapping, with a model of the form

$$o(\underline{i}) = net(\underline{i}) = net_i = \underline{w}^T \underline{i} \qquad \text{where } \underline{w} = \overset{d\times 1}{\begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{pmatrix}} \quad \text{and} \quad \underline{i} = \overset{d\times 1}{\begin{pmatrix} i_1 \\ i_2 \\ \vdots \\ i_d \end{pmatrix}} \tag{3.2}$$

From the viewpoint of geometry, the unit output is formed by projection of the unit input onto the unit weight vector. The weight vector is a specific set of parameters of the unit that models unit interconnection strengths from other sources of stimuli in the network. Thus, the geometric viewpoint indicates that the output (and net activation) of a single linear unit is a measure (see Chapter 2) of the closeness of the input and weight vectors in R^d.

Vector-matrix formulations

The computation of individual-unit net activation may be written

$$net_i = \sum_{j=1}^{d} w_{ij} i_j \tag{3.3}$$

where the jth input to unit i is denoted i_j.[3] This is done to show that a unit input is commonly the output of another unit, in either a feedforward or a recurrent structure. In addition, defining an interconnection matrix W and the input vector $\underline{i}$ as in Equation (3.2),

$$W = [w_{ij}] \tag{3.4}$$

together with a training set H consisting of n S-R pairs, allows the overall network activation to be written as

$$\underline{net} = W\underline{i} \tag{3.5}$$

where

$$\underline{net} = [net_i] \tag{3.6}$$

---

[3] Later this quantity will be written as o_j, to denote the output of *another* unit that provides the jth input to unit i. Even direct inputs to the network, as we see in Chapters 6 and 7, are modeled using (fictitious) "input" units.

Each artificial neuron input has an associated weight indicating the strength of its connection with either an external input or another neuron output. Although the literature is somewhat inconsistent on this topic, we adopt the following convention: *w_{ij} represents the strength of the connection* to *neuron unit i* from *(either) neuron unit j or input j*. Thus, a large positive value of w_{ij} indicates a strongly excitory input from unit or input j to unit i, whereas a large negative weight value may be used to represent a highly inhibitory input.

Single-unit formulation

A single linear unit is perhaps the simplest S-R mapping, with a model of the form

$$o(\underline{i}) = net(\underline{i}) = net_i = \underline{w}^T \underline{i} \qquad \text{where } \underline{w} = \overset{d\times 1}{\begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{pmatrix}} \quad \text{and} \quad \underline{i} = \overset{d\times 1}{\begin{pmatrix} i_1 \\ i_2 \\ \vdots \\ i_d \end{pmatrix}} \tag{3.2}$$

From the viewpoint of geometry, the unit output is formed by projection of the unit input onto the unit weight vector. The weight vector is a specific set of parameters of the unit that models unit interconnection strengths from other sources of stimuli in the network. Thus, the geometric viewpoint indicates that the output (and net activation) of a single linear unit is a measure (see Chapter 2) of the closeness of the input and weight vectors in R^d.

Vector-matrix formulations

The computation of individual-unit net activation may be written

$$net_i = \sum_{j=1}^{d} w_{ij} i_j \tag{3.3}$$

where the jth input to unit i is denoted i_j.[3] This is done to show that a unit input is commonly the output of another unit, in either a feedforward or a recurrent structure. In addition, defining an interconnection matrix W and the input vector $\underline{i}$ as in Equation (3.2),

$$W = [w_{ij}] \tag{3.4}$$

together with a training set H consisting of n S-R pairs, allows the overall network activation to be written as

$$\underline{net} = W\underline{i} \tag{3.5}$$

where

$$\underline{net} = [net_i] \tag{3.6}$$

---

[3]Later this quantity will be written as o_j, to denote the output of *another* unit that provides the jth input to unit i. Even direct inputs to the network, as we see in Chapters 6 and 7, are modeled using (fictitious) "input" units.

Chapter 2 considered formulations of systems of linear equations and associated solutions. The following example is intended to unify those results with the linear unit model and to serve as a prelude to more complex structures and training algorithms.

Case 1. Suppose the training set for the single unit described by Equation (3.2) consists of a *single* S-R mapping of the form

$$H = \{(\underline{i}^p, o^p)\} \tag{3.7}$$

That is, when $\underline{i}^p$ is applied to the unit, the desired response is o^p. Given H, the neuron design problem is then to find the linear unit weights, $\underline{w}$, to achieve this mapping. Equation (3.7) together with Equation (3.2) yield the constraint on $\underline{w}$:

$$o^p = \underline{w}^T \underline{i}^p \tag{3.8}$$

Case 2: Increasing cardinality of H. Suppose H is expanded to include n S-R pairs of the form given in Equation (3.7):

$$H = \{(\underline{i}^p, o^p)\} \quad p = 1, 2, \ldots, n \tag{3.9}$$

It is possible to form the linear vector-matrix equations for the constraints on $\underline{w}$ as

$$\begin{pmatrix} o^1 \\ o^2 \\ \vdots \\ o^n \end{pmatrix} = \begin{pmatrix} (\underline{i}^1)^T \\ (\underline{i}^2)^T \\ \vdots \\ (\underline{i}^n)^T \end{pmatrix} \underline{w} \tag{3.10}$$

3.3.2 Generalizing the Unit Model

Consider now a slightly more general linear unit formulation:

$$o(\underline{i}) = \underline{w}^T \underline{i} + w_o \tag{3.11}$$

where $\underline{i}$ and $\underline{w}$ are as defined in Equation (3.2). Suppose we look at where $o(\underline{i})$ transitions from <0 to >0, i.e., the point

$$o(\underline{i}) = 0 = \underline{w}^T \underline{i} + w_o \tag{3.12}$$

This defines a hyperplane in R^d (see Chapter 2), with the form

$$\underline{w}^T \underline{i} + w_o = 0 \tag{3.13}$$

or

$$\langle \underline{w}, \underline{i} \rangle + w_o = 0 \tag{3.14}$$

It is convenient to reformulate Equation (3.14) so that the appearance is that of a linear model. This relies on the use of *homogeneous coordinate* representations of vectors. Rewriting vectors as

$$\underline{\hat{i}} = \overset{(d+1)\times 1}{\begin{pmatrix} i_1 \\ i_2 \\ \vdots \\ i_d \\ \vdots \\ 1 \end{pmatrix}} \qquad \underline{\hat{w}} = \overset{(d+1)\times 1}{\begin{pmatrix} \underline{w} \\ \vdots \\ w_o \end{pmatrix}} \tag{3.15}$$

we get

$$\underline{\hat{w}}^T \underline{\hat{i}} = 0 = o(\underline{i}) \tag{3.16}$$

Equation (3.11) with $w_o = 0$ is an important computation that may be visualized in many ways:

1. As a weighted sum of inputs, i.e., $o(\underline{i}) = \sum w_j i_j$
2. As a *convolution* or *correlation*[4] of the input $\underline{i}$ with the weights $\underline{w}$
3. As a matched filter
4. As an inner product, as shown previously
5. As a binary classifier, as shown in Figure 3.8

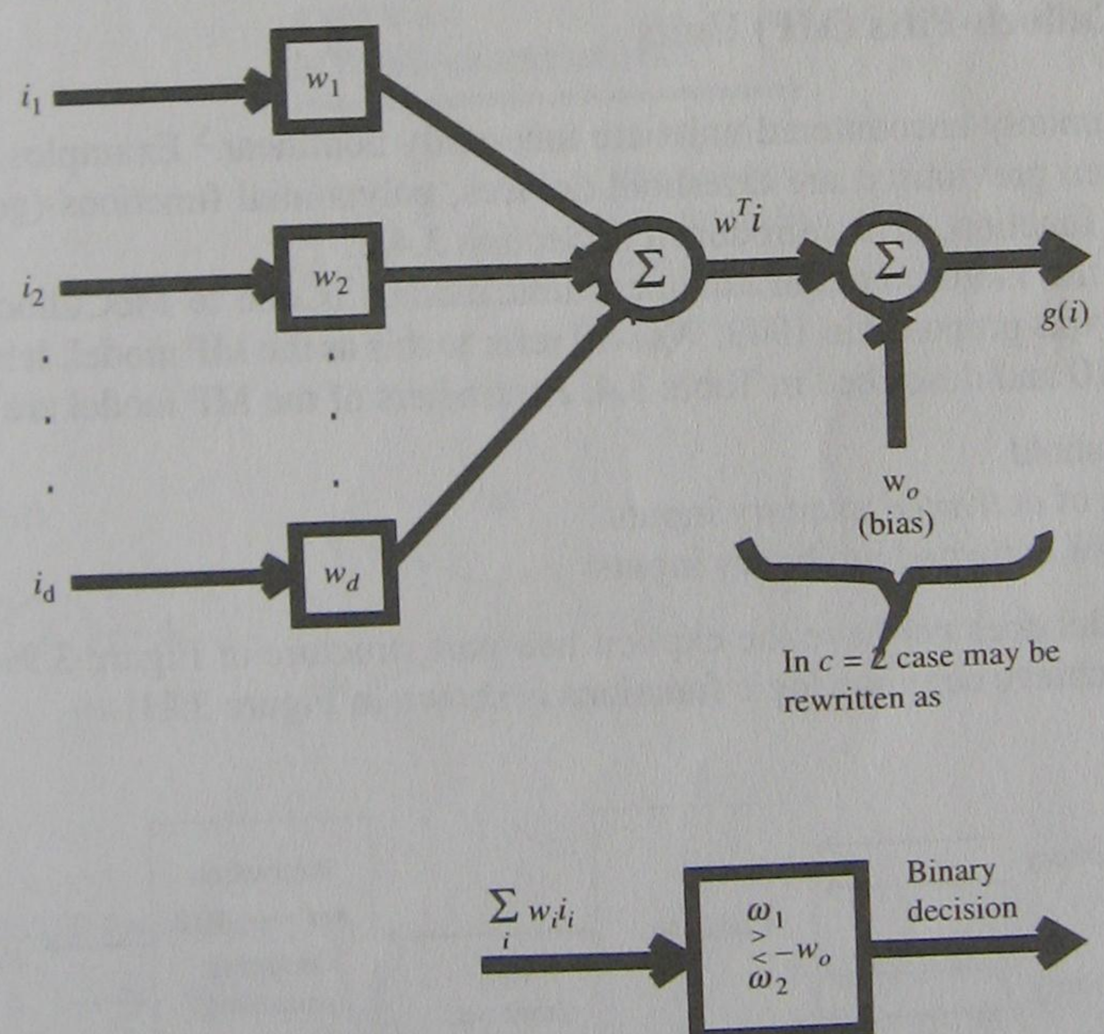


FIGURE 3.8
Using a linear unit as a discriminant function for classification.

---

[4]Note that these are not interchangeable.

3.3.3 Two-Part Unit Models: Activation and Squashing

Many, but not all, of the artificial neural unit models involve two important processes:

1. Forming a *unit net activation* by (somehow) combining (perhaps different classes) of inputs.
2. Mapping this activation value into the artificial unit output. This mapping may be as simple as using the identity function or as complex as using a nonlinear mapping function with memory (dynamics).

There are numerous ways to amalgamate input values to achieve the unit activation value. Common examples are

- Additive: $net = \sum i_1 + i_2 + \cdots + i_d$
- Weighted additive: $net = \sum w_1 i_1 + w_2 i_2 + \cdots + w_d i_d$
- Multiplicative: $net = \pi_i i_i$
- Subtractive
- Polynomial
- Relational, e.g., $net = \max \{i_k\}\ k = 1, 2, \ldots, d$

Figure 3.9 illustrates the generic two-part unit model concept.

3.3.4 McCulloch-Pitts (MP) Units

Certain commonly encountered units are inherently nonlinear.[5] Examples, in addition to those given previously, are threshold devices, polynomial functions (general), and the sigmoid function, to be introduced in Section 3.4.2.

One of the more common nonlinear unit models is due to McCulloch and Pitts [MP43] and was proposed in 1943. We will refer to this as the MP model. It is illustrated in Figure 3.10 and described in Table 3.4. Parameters of the MP model are

T: Threshold
E: Sum of *activated* excitory inputs
I: Sum of *activated* inhibitory inputs

The MP model does not have the explicit two-part structure of Figure 3.9. Use of the MP unit to achieve common logic functions is shown in Figure 3.11.

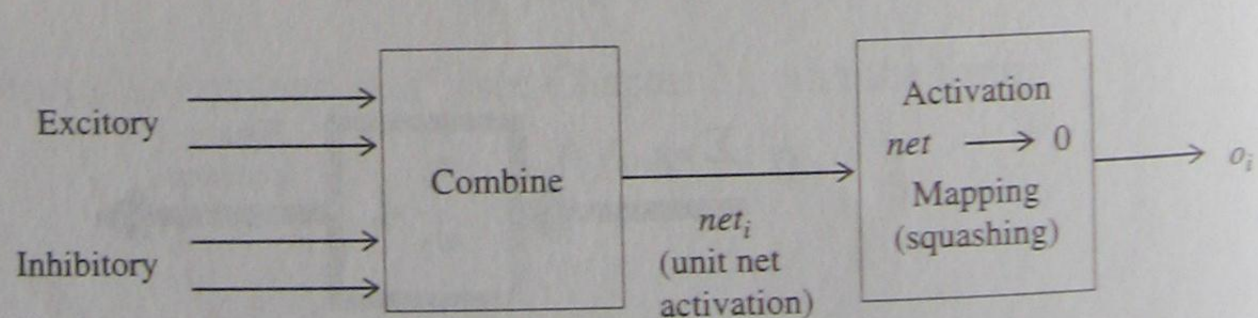


FIGURE 3.9
Two-part model for unit input combination and output formation.

[5]In the I/O sense.

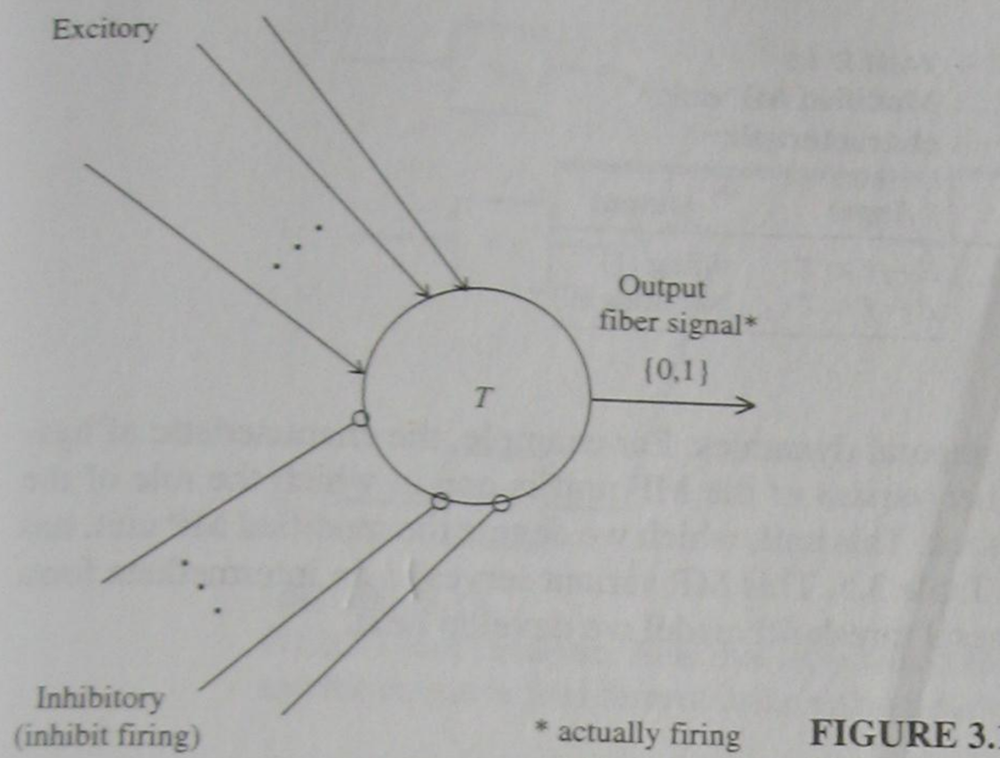


FIGURE 3.10
MP artificial neuron model.

TABLE 3.4
MP unit characteristics

$E \geq T$	$I = 0$	Firing (1)
$E \geq T$	$I > 0$	Not firing (0)
$E < T$	$I = 0$	Not firing (0)
$E < T$	$I > 0$	Not firing (0)

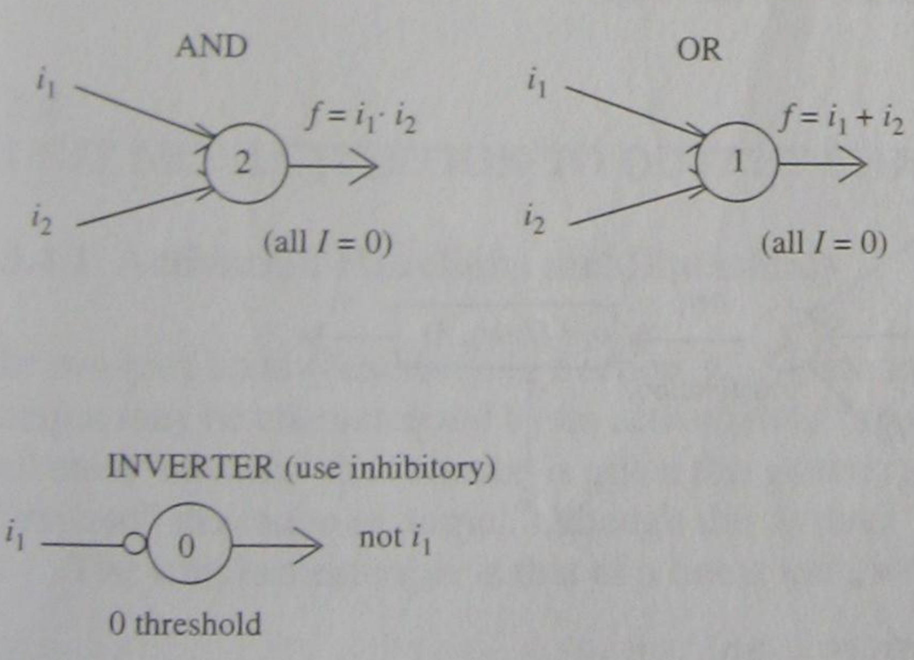


FIGURE 3.11
Achieving simple binary logic functions with the MP unit model.

TABLE 3.5
Modified MP unit characteristic

Input	Output
$E - I \geq T$	Firing (1)
$E - I < T$	Not firing (0)

MP units may exhibit temporal dynamics. For example, the characteristic of hysteresis may be added. Another variant of the MP unit is one in which the role of the inhibitory input(s) is diminished. This unit, which we denote the modified MP unit, has the characteristics shown in Table 3.5. This MP variant serves as an intermediate form between the MP and the general threshold model we develop next.

3.3.5 Threshold Logic with Weighted Linear Input Combination

There are numerous variations of the two-part structure introduced in Figure 3.9. An important and popular neural network building block, which we explore in more detail later, is modeled as a two-part (activation-output mapping) unit, as shown in Figure 3.12. In this case, a linear process determines the unit net activation, and the output is then formed using this quantity.

A specific example of this unit structure is depicted in Figure 3.13. Hereafter, we refer to this unit as the weighted linear input combination with threshold (WLIC-T) structure. Its characteristics are described in Table 3.6. Notice that input weights may take positive or negative values, depending on whether inhibitory or excitory capability is desired. An example [LC67] of the implementation of the Boolean AND using a WLIC-T unit is shown in Figure 3.14.

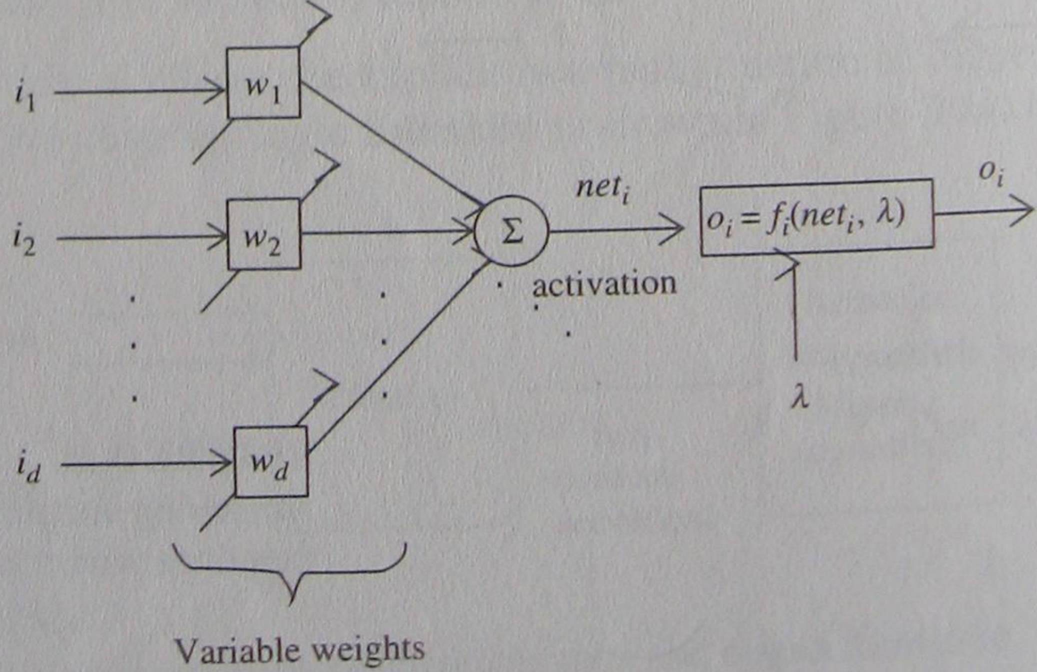


FIGURE 3.12

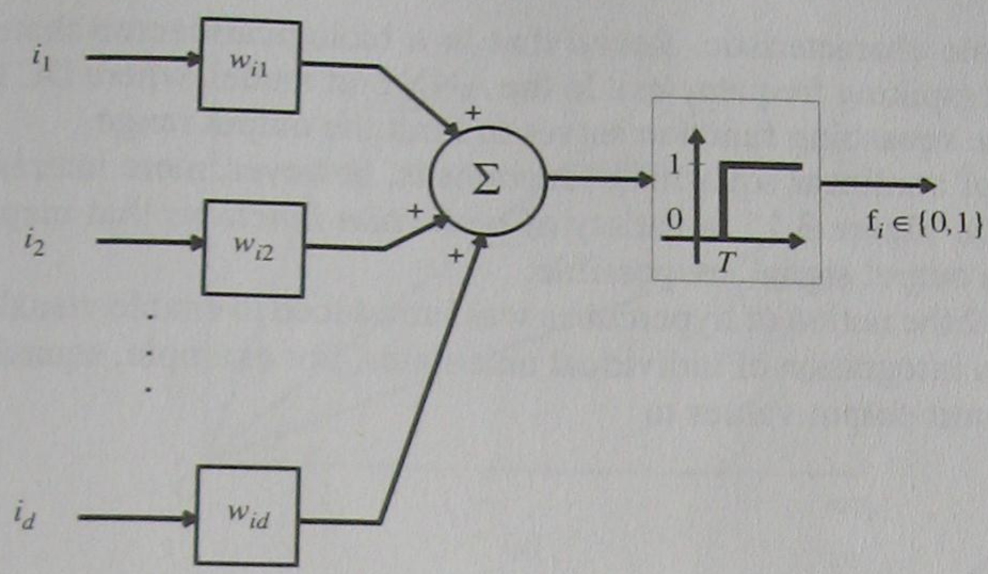


FIGURE 3.13
WLIC-T unit structure. Note that inputs are combined linearly and the output is then formed using a nonlinear device.

TABLE 3.6
WLIC-T unit characteristic

Input characteristic	Output
$\sum_{i=1}^{n} i_i w_i < T$	0
$\sum_{i=1}^{n} i_i w_i \geq T$	1

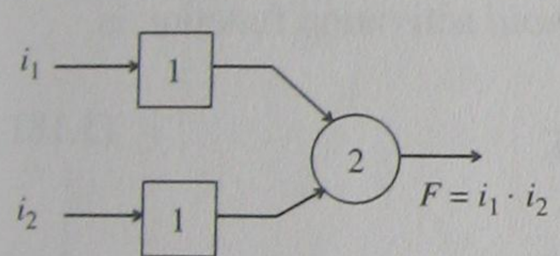


FIGURE 3.14
Application of WLIC-T to achieve logical AND function.

3.4 UNIT NET ACTIVATION TO OUTPUT CHARACTERISTICS

3.4.1 Activation Functions and Squashing

In two-part units (described in Section 3.3.3), the mapping from net unit activation to output may be characterized by an activation or "squashing" function. Even though not all units "squash," the function is given this generic name. An activation unit could also "expand" the range of output, although this is rare.

The simplest example is that of a linear unit, where

$$o_i = f(net_i) = net_i \tag{3.17}$$

In this case, the activation function is the identity mapping.

In activation functions that implement input-to-output compression or squashing, the range of the function is less than that of the domain. There is some physical basis for this desirable characteristic. Recall that in a biological neuron there is a limited range of output (spiking frequencies). In the ANN unit model, where DC levels replace frequencies, the squashing function serves to limit the output range.

The class of nonlinear squashing functions is, however, more interesting and useful. As shown in Figure 3.15, a variety of *activation functions* that map neuron input activation to an output signal are possible.

In Chapter 2 the notion of hypercubes was introduced to enable visualization of network state as an integration of individual unit states. For example, squashing functions may constrain unit output values to

- $o_i \in [0, 1]$
- $o_i \in (0, 1)$
- $o_i \in \{0, 1\}$
- $o_i \in [-1, 1]$
- $o_i \in (-1, 1)$
- $o_i \in \{-1, 1\}$

Each yields an interpretation of the network state based on the corresponding hypercube.

3.4.2 The Sigmoid (Logistic) Squashing Function

The *logistic* function has a rich history of application as a cumulative distribution function in demographic studies and in modeling growth functions [Bal92]. The particular functional form that is often used for the *logistic* or *sigmoid* activation function is

$$o_i = f(net_i) = \frac{1}{1 + e^{-net_i}} \tag{3.18}$$

which yields $o_i \in [0, 1]$. This is shown is Figure 3.16.

There are several reasons the sigmoid is so important and popular:

1. It squashes.
2. It is semilinear.[6] This influences its use with certain training approaches.
3. It is expressible in closed form.
4. Modifications or extensions lead to or relate to other squashing functions.
5. The derivative of the sigmoid with respect to net_i is very easy to form.
6. It has a biological basis. The average firing frequency of biological neurons, as a function of excitation, follows a sigmoidal characteristic.

Sigmoid derivative

The derivative of the logistic function is

$$f'(x) = \frac{e^{-x}}{(1 + e^{-x})^2} \tag{3.19}$$

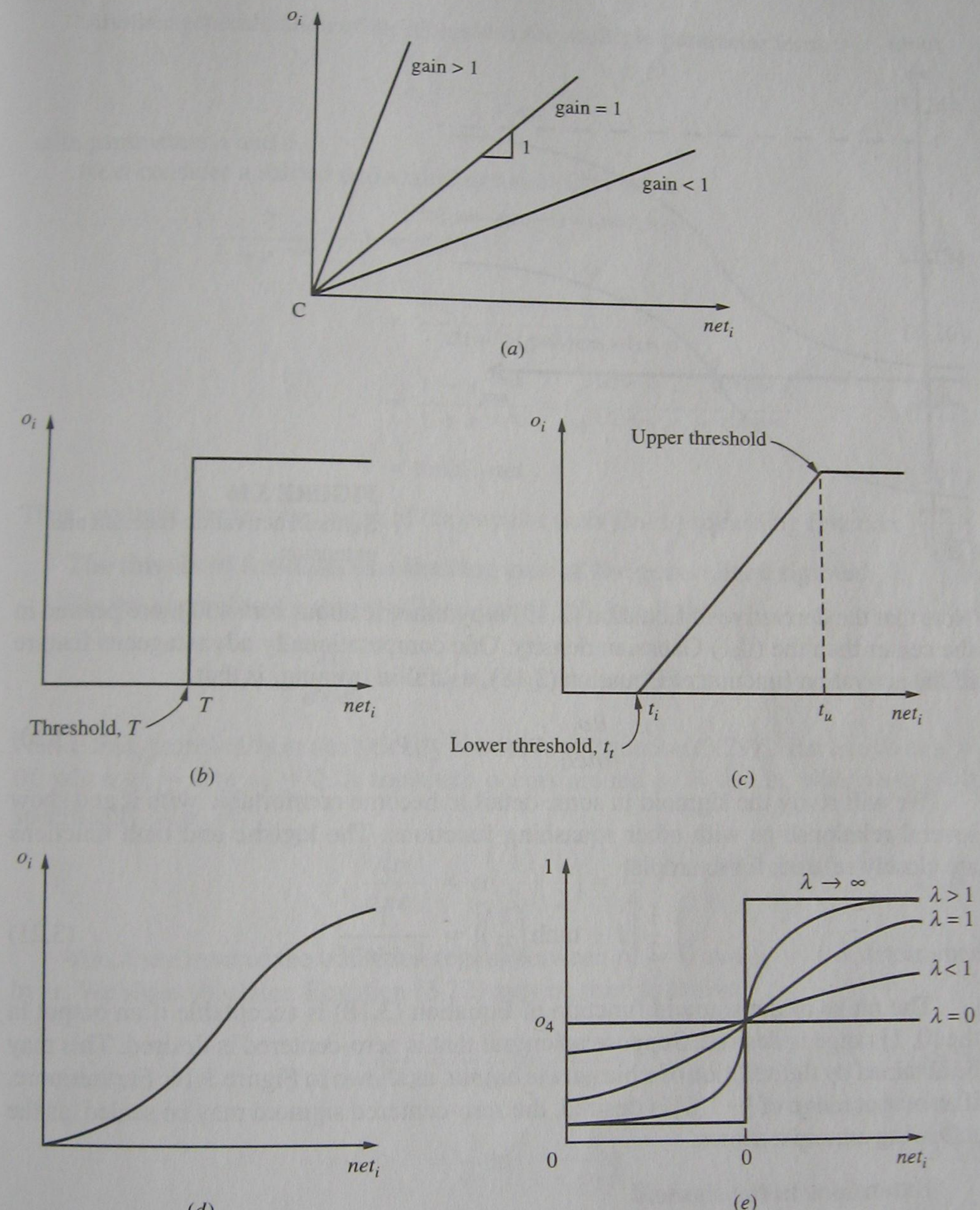


FIGURE 3.15
Common squashing functions.

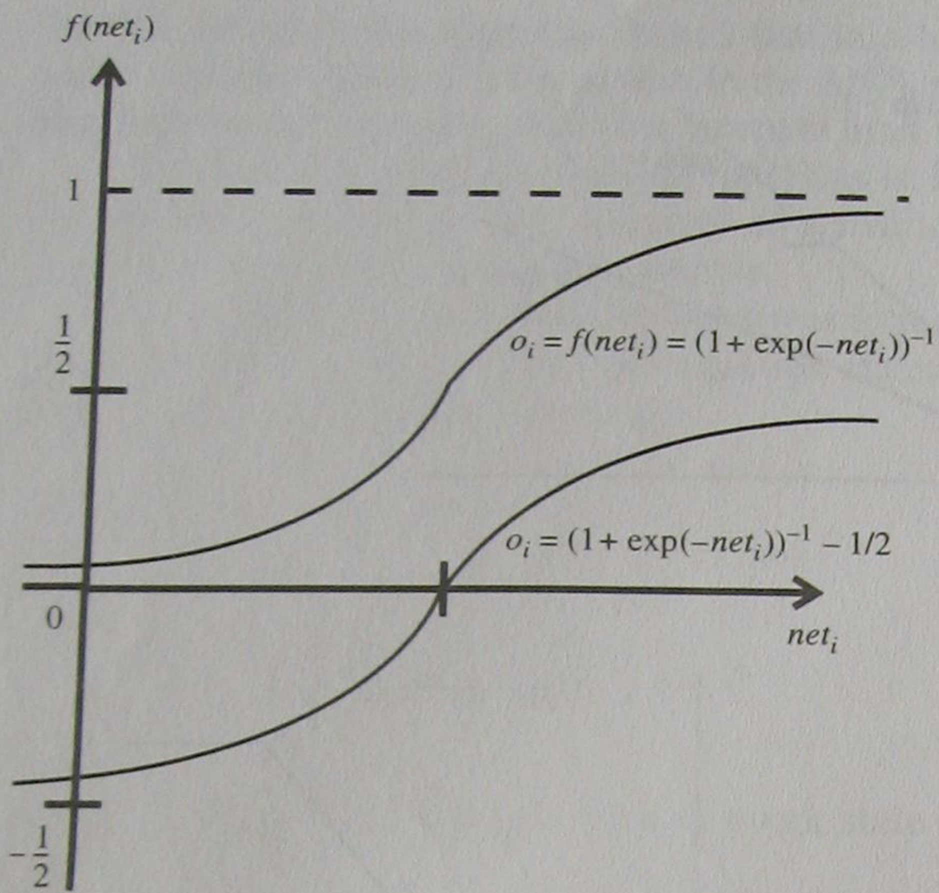


FIGURE 3.16
Sigmoid activation function and extension.

Note that the derivative in Equation (3.19) is symmetric about zero and more peaked in the center than the (0,1) Gaussian density. One computationally advantageous feature of the activation function of Equation (3.18), useful in training, is that

$$\frac{\partial o_i}{\partial net_i} = o_i \cdot (1 - o_i) \tag{3.20}$$

We will study the sigmoid in some detail to become comfortable with it and show several relationships with other squashing functions. The logistic and tanh functions are closely related; for example,

$$\frac{1}{2}\left[1 + \tanh\left(\frac{x}{2}\right)\right] = \frac{1}{1 + e^{-x}} \tag{3.21}$$

The range of the sigmoid function of Equation (3.18) is acceptable if an output in the [0, 1] range is desired. Suppose an output that is zero-centered is desired. This may be obtained by the addition of a bias *at the output*, as shown in Figure 3.16. Furthermore, if an output range of $[-1, 1]$ is desired, the zero-centered sigmoid may be scaled, as the following example shows.

Extensions to the sigmoid

The sigmoid or logistic function of Equation (3.18) may be generalized in several ways. The first is via the addition of a "gain," namely,

$$o_i = \frac{1}{1 + e^{-\alpha net_i}} \tag{3.22}$$

where α is the gain parameter. In this case, the reader should verify that

$$\frac{\partial o_i}{\partial net_i} = \alpha o_i \cdot (1 - o_i) \tag{3.23}$$

Another generalization of the sigmoid is the multiple-parameter form:

$$o_i = \frac{1}{1 + ae^{-bnet_i}} \tag{3.24}$$

with parameters a and b.

Next consider a shifted and scaled sigmoid of the form

$$\frac{2}{1 + e^{-net_i}} - 1 = f_m(net_i) \tag{3.25}$$

$$= \frac{-1 - e^{-net_i} + 2}{1 + e^{-net_i}} \tag{3.26}$$

$$= \frac{1 - e^{-net_i}}{1 + e^{-net_i}} = \frac{e^{(1/2)net_i} - e^{-(1/2)net_i}}{e^{(1/2)net_i} + e^{-(1/2)net_i}} \tag{3.27}$$

$$= \tanh(\tfrac{1}{2}net_i) \tag{3.28}$$

Thus, we have shown one origin of the popular $\tanh(\beta net_i)$ squashing function.

The threshold function as a limiting case of the generalized sigmoid

For the generalized sigmoid of Equation (3.22), recall that

$$\frac{do_i}{dnet_i} = \alpha o_i(1 - o_i) = \alpha o_i - \alpha o_i^2 \tag{3.29}$$

Notice that, graphically or analytically, the curve of Equation (3.29) is flat ($\alpha o_i / \alpha net_i = 0$), where $o_i \approx 1$ or $o_i \approx 0$. A transition occurs around $o_i = \frac{1}{2}$, i.e., where $net_i = 0$. specifically, at $o_i = \frac{1}{2}$,

$$\frac{do_i}{dnet_i} = \alpha\left(\frac{1}{2}\right)\left(\frac{1}{2}\right) = \frac{\alpha}{4} \tag{3.30}$$

Also, the *extent* of the transition region between $o_i = 0$ and $o_i = 1$ is determined by α. We show this later. Equation (3.22) may be used to derive

$$net_i = -\frac{1}{\alpha}\ln\left(\frac{1 - o_i}{o_i}\right) \tag{3.31}$$

$$= \frac{1}{\alpha}\ln\left(\frac{o_i}{1 - o_i}\right) \tag{3.32}$$

Suppose we define the transition region to be the extent of net_i values such that $0.37 \leq o_i \leq 0.63$. Notice that this is an interval of ± 0.13 on either side of $o_i = \frac{1}{2}$ or $net_i = 0$. Since, for $\alpha = 1$,

$$net_i|_{o_i=0.37} = -0.532 \tag{3.33}$$

and

$$net_i|_{o_i=0.63} = 0.532 \tag{3.34}$$

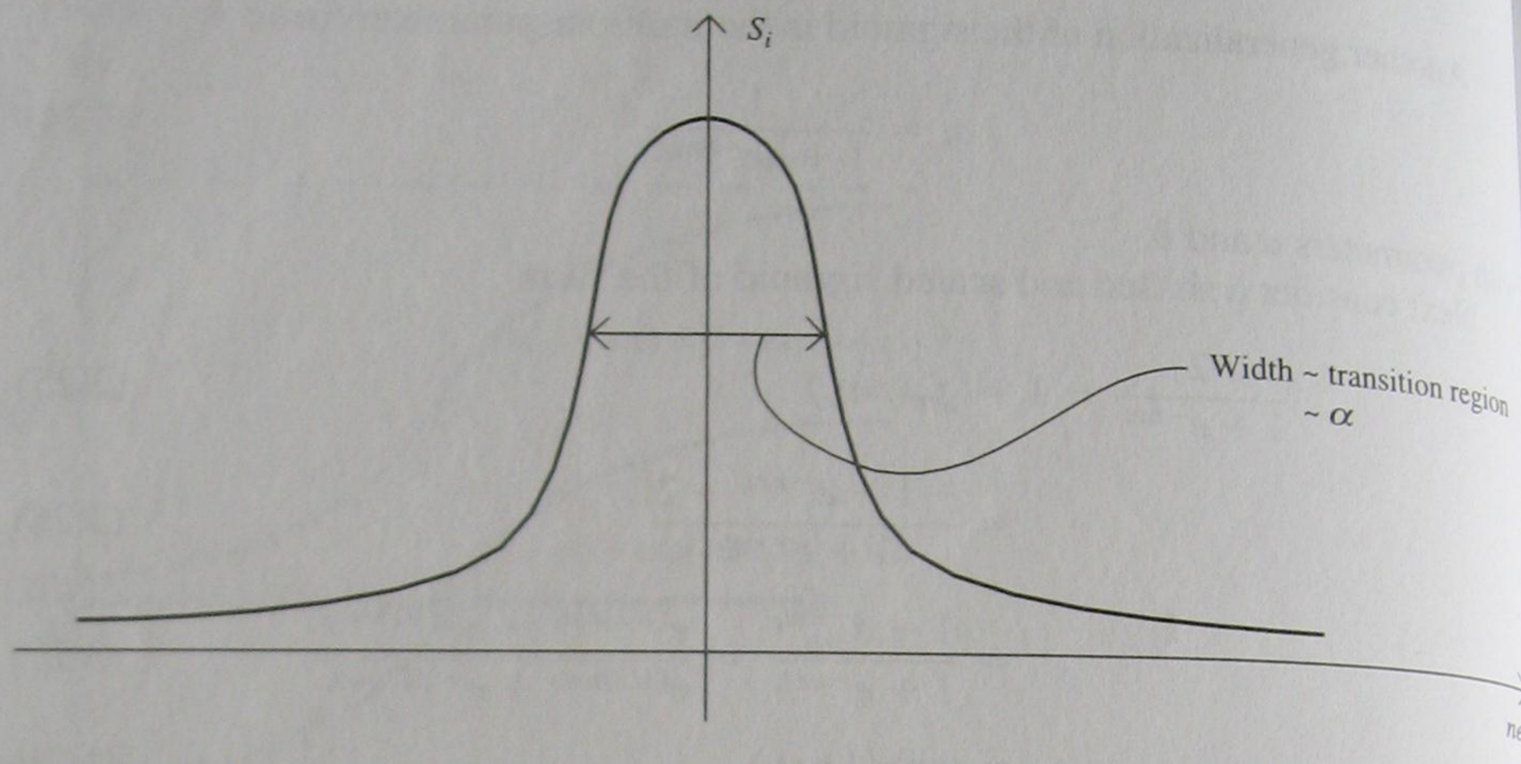


FIGURE 3.17
Generalized sigmoid derivative, s_i.

the transition region $t_r(\alpha = 1) \approx 2(0.532) = 1.064$ (net_i units) wide. Repeating the derivation for $\alpha = 10$, $t_r(\alpha = 10) = 0.106$. Clearly,

$$t_r(\alpha) \sim \frac{1}{\alpha} \tag{3.35}$$

and as α gets large, $t_r(\alpha) \rightarrow 0$. If we plot the quantity

$$s_i(net_i) = \frac{d[o_i(net_i)]}{dnet_i} \tag{3.36}$$

versus net_i, as shown in Figure 3.17, we observe two things:

1. s_i has a single peak centered around $net_i = 0$ since

$$s_i = \alpha o_i - \alpha o_i^2 \tag{3.37}$$

2. $$\frac{ds_i}{do_i} = \alpha - 2\alpha o_i = 0 \tag{3.38}$$

   yields $o_i = \frac{1}{2}$, corresponding to $net_i = 0$. Thus, α controls both $t_r(\alpha)$ and the peak value of s_i.

   - If $\alpha \rightarrow 0$, s_i has a small peak and is spread over a wide region.
   - $\alpha \rightarrow \infty$ yields a narrow (ideally zero) transition region with a large (ideally infinite) peak.

This similarity with another well-known function, the Dirac delta function, leads us to conclude that

$$\lim_{\alpha \rightarrow \infty} \{s_i(net_i)\} = \delta(net_i) \tag{3.39}$$

The fact that $s_i(net_i) \rightarrow \delta(net_i)$ as $\alpha \rightarrow \infty$ yields the corresponding characteristic that the generalized sigmoid approaches the unit step. For this reason, we may approximate the threshold characteristic ($t = 0$) of the step by a generalized sigmoid with large gain.

Linearization of the sigmoid

Chapter 2 showed the use of Taylor series expansions to generate linear models. Given

$$o_i = \frac{1}{1 + e^{-net_i}} \tag{3.40}$$

recall

$$f(net_i + \Delta net_i) = f(net_i) + \frac{df(net_i)}{dnet_i}\Delta net_i + \text{ higher-order terms} \tag{3.41}$$

where $f(net_i)$ and the derivative are evaluated at $net_i = 0$. The linear version of Equation (3.41) ignores the higher-order terms.

Case 1: net$_i$ = 0. The reader is left to show that applying Equation (3.41) to Equation (3.40) yields

$$f(\Delta net_i) \approx f(0) + o_i(1 - o_i)|_{net_i=0}\Delta x = \frac{1}{2} + \frac{\Delta net_i}{4} \tag{3.42}$$

The reader should check the validity of this linear approximation in regions close to $net_i = 0$ as well as others.

Case 2: net$_i$ = 2. Here

$$\begin{aligned} f(2 + \Delta net_i) &\approx f(2) + o_i(1 - o_i)|_{net_i=2}\Delta net_i \\ &= 0.881 + 0.167\Delta net_i \end{aligned} \tag{3.43}$$

The use of linearized models leads to fitting line segments to the sigmoid curve in local regions.

3.4.3 Other Squashing Functions

There exist many alternative (and sometimes related) squashing functions. A sample piecewise linear activation function is

$$o_i = \begin{cases} -1 & net_i < -1 \\ net_i & |\, net_i \,| \leq 1 \\ +1 & net_i > 1 \end{cases} \tag{3.44}$$

This function, although composed of linear segments, is nonlinear.

A piecewise linear, threshold-based squashing function is

$$o_i = \begin{cases} 1 & \alpha net_i \geq 1 \\ \alpha net_i & 0 \leq \alpha net_i < 1 \\ 0 & \alpha net_i < 0 \end{cases} \tag{3.45}$$

The "slow" sigmoid is

$$o_i = \frac{net_i}{1 + |net_i|} \tag{3.46}$$

Note that the range of this function is the interval $[-1, 1]$.

The hyperbolic tangent is defined as

$$\tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3.47}$$

and gives rise to another popular semilinear activation function. This extension of the sigmoid, the hyperbolic tangent function, was derived in Section (3.4.2).

$$o_i = \tanh(\beta net_i) \tag{3.48}$$

$$= \frac{e^{\beta net_i} - e^{-\beta net_i}}{e^{\beta net_i} + e^{-\beta net_i}} \tag{3.49}$$

A particularly interesting class of activation functions, introduced in Sections 3.3.4 and 3.3.5, are *bilevel* mappings or thresholding units, for example,

$$o_i = f(net_i) = \begin{cases} 1 & net_i \geq 0 \\ 0 & net_i < 0 \end{cases} \tag{3.50}$$

Figure 3.13 shows this unit characteristic using a more general threshold, T. Alternatively,

$$o_i = f(net_i) = \begin{cases} 1 & net_i \geq 0 \\ -1 & net_i < 0 \end{cases} \tag{3.51}$$

Thresholding is a limiting case of the variable-gain sigmoidal unit characteristic of Equation (3.18). This is shown in Figure 3.15. Thresholding units may be used to compute Boolean functions.

Two of the most important limitations of threshold (hardlimiter) units are

1. Lack of invertibility of the activation-output mapping.
2. The inability to compute a derivative of the mapping.

These characteristics will become a key issue in the design of training algorithms.

The basis for forming the net unit activation in Equation (3.3) is an inner-product operation. In the case where the activation-output mapping is linear, e.g., Equation (3.17), we observe that the neuron implements a linear discriminant function. Alternatively, input space R^d is partitioned into two half planes, determined by the unit weights (Chapter 2). Thresholding units, defined in Equations (3.50) and (3.51), directly implement this partitioning. In a thresholding unit, the effect of weight w_{ij} is to stretch or contract the jth axis of the hyperspace. Thus, a single thresholding unit can directly implement the decision boundary to a linearly separable problem.

Signum or sign functions are used in Chapter 4 and are related to the threshold unit. A common definition is

$$o_i = f_i(net_i) = \mathrm{sgn}(net_i) = \begin{cases} +1 & net_i > 0 \\ 0 & net_i = 0 \\ -1 & net_i < 0 \end{cases} \tag{3.52}$$

The signum function, as defined in Equation (3.52), does not produce a binary (two-level) output.

The concept of using combinations of min and max functions generates families of other squashing functions, for example,

$$o_i = \min\{1, e^{\alpha net_i}\} \tag{3.53}$$

$$o_i = \max\{0, 1 - e^{-\alpha net_i}\} \tag{3.54}$$

and

$$o_i = \max\left\{0, \frac{(net_i)^n}{c + (net_i)^n}\right\} \tag{3.55}$$

An example of a nonsaturating activation function is

$$f(net_i) = \begin{cases} \log(1 + net_i) & net_i > 0 \\ -\log(1 - net_i) & net_i < 0 \end{cases} \tag{3.56}$$

3.4.4 Exceptions to the Two-Part Model

Many unit models do not allow visualization in two parts. These include the MP model. In fact, the attempted visualization of "weights" for an MP unit is meaningless. In addition, many unit models, although they may be described as two-part processes, do not use WLIC-T for the formation of net activation. An example of such a unit model is the polynomial unit, used in so-called higher-order neural networks. Polynomial units form their outputs by computing some polynomial function of their inputs. A multiplicative unit, where

$$o_i = \prod_{i=1}^{d} i_1 i_2 \ldots i_d \tag{3.57}$$

is another case where a linear input stage is not used. A final example is the *radial basis unit* model of Chapter 10.

Finally, there is an active area of research involving "weightless" neural nets. Weightless units use RAM lookup tables and derivatives of these to construct artificial neurons [Gur92].

There is an important class of units in which output is computed based on the minimum or maximum of input values. Although this unit mapping is obviously expressible in closed form, it is not possible to differentiate the output with respect to the input vector,[7] and therefore approximations may be employed. Two common approximations

[7] A characteristic that will become important in Chapter 6.

are the "soft" min and max, defined as

$$\text{softmin}\,(i_1, i_2, \ldots, i_d) = \frac{i_1 e^{-ki_1} + i_2 e^{-ki_2} + \cdots + i_d e^{-ki_d}}{e^{-ki_1} + e^{-ki_2} + \cdots + e^{-ki_d}} \tag{3.58}$$

$$\text{softmax}\,(i_1, i_2, \ldots, i_d) = \frac{i_1 e^{ki_1} + i_2 e^{ki_2} + \cdots + i_d e^{ki_d}}{e^{ki_1} + e^{ki_2} + \cdots + e^{ki_d}} \tag{3.59}$$

where k is a positive constant that controls the "crispness" of the approximation. Although it is difficult to pinpoint the origins of these approximations, one source is the "maxentropy" approach of E. T. Jaynes used for the estimation of probabilities and related functions [Jay89]. These approximations are used in the problems at the end of this chapter and in Chapter 11 to implement fuzzy logic functions.

3.4.5 "Memory" or Individual Unit Activation Dynamics

Most of the artificial units we have considered up to this point provide static I/O mappings. Previous unit states have no influence on subsequent mappings. An exception is the MP unit with hysteresis. Unit temporal dynamics may include I/O lag, delay, memory, and other effects.

A typical and more general model that incorporates unit dynamics might be

$$\frac{d(net_i(t))}{dt} = f_a(net_i(t)) + \alpha_i net_i(t) \tag{3.60}$$

where $net_i(t)$ is the activation of the ith neuron and α_i is a time constant. This constrains the time change of individual unit states and enables a local "memory." For a discrete-time model, similar difference equations may be derived.

3.5 ARTIFICIAL UNIT MODEL EXTENSIONS

3.5.1 Adding (an Optional) Bias to the Artificial Neuron Model

Another neuron parameter is the (optional) bias or offset input into the unit. Although this could be achieved simply by adding a constant input with an appropriate weight, often the bias is considered separately. Biases may be used, for example, to selectively inhibit the activity of certain neurons.

Recalling the sigmoidal activation function

$$f(net_j) = \frac{1}{1 + e^{-net_j}} \tag{3.61}$$

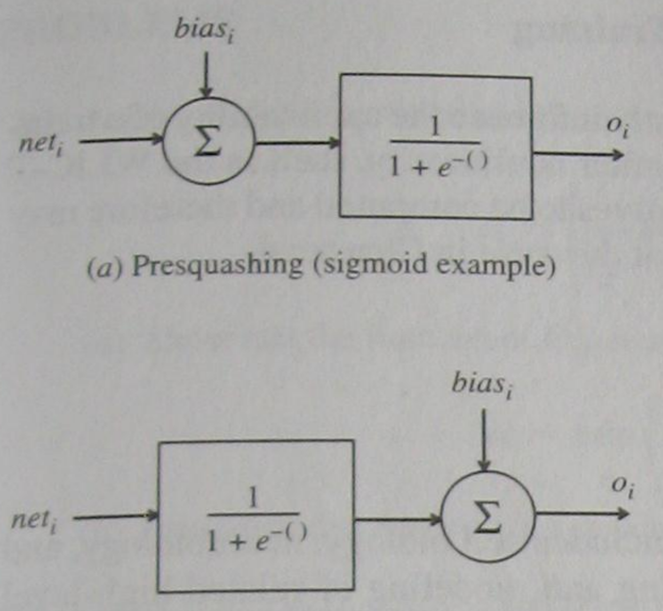
we observe that

$$0 \leq f(net_j) \leq 1 \tag{3.62}$$

and with no activation, i.e.,

$$net_i = 0 \tag{3.63}$$

$$f(0) = \tfrac{1}{2} \tag{3.64}$$



(*a*) Presquashing (sigmoid example)

(*b*) Postsquashing (sigmoid example)

FIGURE 3.18
Adding a bias input: two approaches.

We may wish to bias this unit such that $f(0)$ is another value. As we show later, this bias may also be adjusted as part of the network training. A simple model for the unit with bias is to modify net_j such that

$$net_j = \sum_i w_{ji} o_i + bias_j \tag{3.65}$$

In contrast with the sigmoid, tanh units have an I/O characteristic that is symmetric with respect to the origin.

Note that the bias as just introduced is not the only possible way to incorporate this parameter into the unit characteristic. In fact, it may not be the most effective for some applications, as shown in Figure 3.18.

3.5.2 Inhibitory Inputs

It is often desirable to have neuron unit inputs that serve to inhibit the unit's activation. An example of this utility is found in *competitive learning* (Chapter 9). This characteristic may be achieved in several ways. As shown in Figure 3.13, negative values of w_{ij} that are large in magnitude, due to the summation in Equation (3.3), yield strong inhibitory characteristics. Alternatively, a more severe form of inhibition may be achieved through a nonlinear activation-output model, as shown in Figure 3.10.

3.5.3 Individual Unit Dynamics versus Network Dynamics

Care must be taken to distinguish the aforementioned *unit dynamics* from the dynamics of the neural network. In many cases (e.g., Hopfield nets) the overall network is a large, highly interconnected system of nonlinear elements with feedback. Putting aside stability concerns, such a network, when started in some state, will typically display time-varying behavior or dynamics. These may be described using either differential or difference equations. This is the case *even if the individual units are static input/output mappings, i.e., have no individual dynamic behavior.*

3.5.4 Activation Functions and Network Training

The choice of activation function may significantly influence the applicability of a training algorithm. For example, units with a hardlimiter nonlinearity, such as the WLIC-T of Section 3.3.5, do not allow certain unit derivatives to be computed and therefore may not allow gradient descent solutions.[8] We look at this case in Chapter 4.

with linear A.F.

3.6 BIBLIOGRAPHY

The potential complexity of biological models includes cell biology, neurobiology, and organic chemistry. In addition, the understanding and modeling of related high-level cognitive processes may require more than a casual background in psychology. Fortunately, there are a number of references useful to those without an extensive life science background. Relatively simple biological models are treated in a very readable manner in Chapter 3 of [Kan91]. Synapses, from both biological and functional viewpoints, are treated in [KP87]. A unified view of mind and brain is treated in [Chu86]. The processes of learning, memorization, neural connectivity, and elementary reasoning are covered very well in [Val94].

REFERENCES

[Bal92] N. Balakrishnan. *Handbook of the Logistic Distribution*. Marcel Dekker, New York, 1992.

[Chu86] Patricia S. Churchill. *Neurophilosophy: Toward a Unified Science of the Mind-Brain*. MIT Press, Cambridge, MA, 1986.

[CS93] P. S. Churchland and T. J. Sejnowski. *The Computational Brain*. MIT Press, Cambridge, MA, 1993.

[Gur92] K. N. Gurney. Weighted nodes and ram-nets: A unified approach. *Journal of Intelligent Systems*, 2:155–186, 1992.

[Jay89] Edwin T. Jaynes. *Papers on Probability, Statistics and Statistical Physics*. Kluwer Academic Publishers, Norwell, MA, 1989.

[Kan91] Eric R. Kandel. Nerve cells and behavior. In Eric R. Kandel, James H. Schwartz, and Thomas M. Jessell, eds. *Principles of Neural Science*, 3rd ed. Elsevier, New York, 1991.

[KP87] C. Koch and T. Poggio. Biophysics of computation: Neurons, synapses and membranes. In G. M. Edelman, ed. *Synaptic Function*. John Wiley & Sons, New York, 1987.

[LC67] P. M. Lewis II and C. L. Coates. *Threshold Logic*. John Wiley & Sons, New York, 1967.

[MP43] W. S. McCulloch and W. Pitts. A logical calculus of the ideas imminent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.

[Val94] Leslie G. Valiant. *Circuits of the Mind*. Oxford University Press, New York, 1994.

[8] At least not directly.

PROBLEMS

3.1. A piecewise linear, threshold-based squashing function is

$$o_i = \begin{cases} 1 & \alpha net_i \geq 1 \\ \alpha net_i & 0 \leq \alpha net_i < 1 \\ 0 & \alpha net_i < 0 \end{cases} \tag{3.66}$$

(*a*) Show that the function of Equation (3.66) may be written as

$$o_i = \min\left\{1, \max\{0, \alpha net_i\}\right\} \tag{3.67}$$

(*b*) Show that this function is a nondecreasing function of $net_i (\alpha > o)$.

3.2. (*a*) Develop three-input AND, OR, NOT, and XOR logic functions using the WLIC-T unit whose characteristics are shown in Table 3.6 (repeated as Table P3.2) as the basic building block.

(*b*) Relate the MP implementations of these logic functions with those of the linear threshold model.

TABLE P3.2
WLIC-T unit characteristic

Input characteristic	Output
$\sum_{i=1}^{n} i_i w_i < T$	0
$\sum_{i=1}^{n} i_i w_i \geq T$	1

3.3. Plot each of the following squashing functions[9] on the same scale and discuss relevant characteristics such as (but not limited to)

- Points of inflection
- Ease of computing the derivative
- Flexibility of adjustment of saturation points, intercept, etc.
- Symmetry with respect to axes

Use different parameters and plot the functions on the same scale for comparison.

(*a*) $o_i = \min\{1, e^{\alpha net_i}\}, \quad \alpha > 0$
(*b*) $o_i = (1 + ae^{-\beta net_i})^{-1}$
(*c*) $o_i = \tanh(\beta net_i)$
(*d*) $o_i = \min\{1, \max\{0, \alpha net_i\}\}$
(*e*) $o_i = \max\{0, 1 - e^{-\alpha net_i}\}$
(*f*) $o_i = \max\left\{0, \dfrac{net_i^n}{c + net_i^n}\right\}$
(*g*) $o_i = net_i/(1 + |net_i|)$

[9]Choose relevant parameter values where necessary.

3.4. Show, using the definition in Table 3.4 (repeated here as Table P3.4), that the MP neuron is nonlinear in the input/output sense.

TABLE P3.4
MP neural unit characteristics

$E \geq T$	$I = 0$	Firing (1)
$E \geq T$	$I > 0$	Not firing (0)
$E < T$	$I = 0$	Not firing (0)
$E < T$	$I > 0$	Not firing (0)

3.5. Can the MP model be put into a two-part structure as in Figure 3.13?

3.6. Show that the AND logic function is nonlinear in the I/O sense. Is the OR function linear in the I/O sense?

3.7. Can you think of situations where the "fail-safe" role of the inhibitory input(s) is useful? ("No" is not a satisfactory answer.)

3.8. A useful unit is the "majority cell," where the cell fires if a majority of inputs are activated. Using the MP cell (Table P3.4), show how this may be achieved. Can you think of an application for such a cell?

3.9. Show how the modified MP characteristic of Table 3.5 (repeated as Table P3.9) may be achieved with the WLIC-T unit.

TABLE P3.9
Modified MP unit characteristic

Input	Output
$E - I \geq T$	Firing (1)
$E - I < T$	Not firing (0)

3.10. Repeat the extension of the sigmoid into the tanh function of Section 3.4.2 beginning with the generalized sigmoid: $o_i = f(net_i) = 1/1 + e^{-\alpha net_i}$.

3.11. The sigmoid derivative[10] is given by

Sac.

$$s_i(net_i) = \frac{d[o_i(net_i)]}{dnet_i} \quad (3.6\ldots)$$

The Dirac delta function, $\delta(x)$, has the property of unit area, i.e.,

$$\int \delta(x)\,dx = 1.0 \quad (3.6\ldots)$$

Does this hold for the sigmoid derivative $s_i(net_i)$?

[10]Generalized sigmoid with $\alpha = 1$.

3.12. Compare the derivative of the sigmoid activation function, presented in Section 3.4.2, with the (0, 1) Gaussian density function.

3.13. Verify, through simulation, the behavior of the softmin and softmax approximations.

3.14. Consider the "slow sigmoid" function, i.e.,

$$o_i = \frac{net_i}{1 + |\, net_i \,|} \tag{3.70}$$

(*a*) Plot o_i as a function of net_i. Why do you suppose the connotation "slow" is used?
(*b*) Compare this function with the generalized sigmoid.
(*c*) How would you compute $\partial o_i / \partial net_i$?

3.15. Consider an artificial neural unit with the following input/output characteristic:

$$o_i = \max\{i_1, i_2, \ldots, i_d\} \tag{3.71}$$

(*a*) Is this unit linear?
(*b*) How would you compute $\partial o_i / \partial i_i$ for the unit of Equation (3.71)?
(*c*) Show how the unit of Equation (3.71) could be used to implement simple logic functions such as
 (i) AND
 (ii) OR

3.16. For the tanh activation function,

$$o_i = \tanh(\beta\, net_i)$$

(*a*) Compute $\partial o_i / \partial net_i$.
(*b*) Relate and compare this to the generalized sigmoid derivative.
(*c*) Verify the following:

$$\frac{d}{dx}\{\tanh(x)\} = \operatorname{sech}^2(x) = 1 - \tanh^2(x) \tag{3.72}$$

3.17. We have shown a linearization of the sigmoid in Section 3.4.2. This problem considers several alternatives. Recall

$$e^{-x} = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \cdots \tag{3.73}$$

and

$$\frac{1}{1-x} = 1 + x + x^2 + \cdots \tag{3.74}$$

Can you use these approximations to develop alternative approximations for the basic sigmoid function of Equation (3.40)?

3.18. Compute f_i' for the nonsaturating activation function:

$$f(net_i) = \begin{cases} \log(1 + net_i) & net_i > 0 \\ -\log(1 - net_i) & net_i < 0 \end{cases} \tag{3.75}$$

3.19. Verify the following statement: The sigmoid function $[f(x) = 1/(1 + e^{-x})]$ is a smoothed version of $1/2[\mathrm{sgn}(x) + 1]$.

3.20. (*a*) For Equation 3.40, compute $d^2 f/dnet^2$.

(*b*) Referring to Section 3.4.2, repeat the derivation of the linear models using second-order expansions; i.e., include $d^2 f(net_i)/dnet_i^2$ terms. Compare the validity of the second-order model around the point of linearization as well as the region over which the model is valid with that of the linear version.