these learning tasks are all problems of learning a *mapping* from (possibly noisy) examples of the mapping. Without the imposition of prior knowledge, each of the tasks is in fact *ill posed* in the sense of nonuniqueness of possible solution mappings. One method of making the solution well posed is to use the theory of regularization as described in Chapter 5.

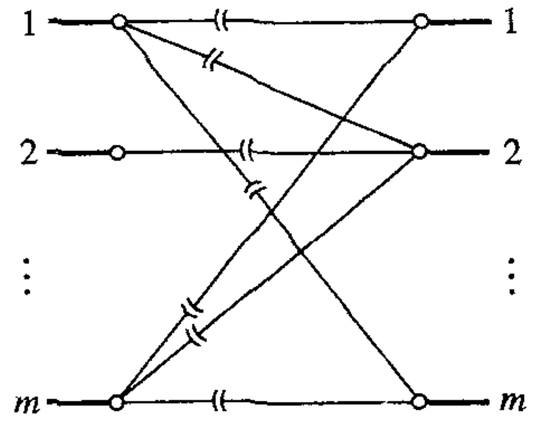
2.11 MEMORY

Discussion of learning tasks, particularly the task of pattern association, leads us naturally to think about *memory*. In a neurobiological context, memory refers to the relatively enduring neural alterations induced by the interaction of an organism with its environment (Teyler, 1986). Without such a change there can be no memory. Furthermore, for the memory to be useful it must be accessible to the nervous system in order to influence future behavior. However, an activity pattern must initially be stored in memory through a *learning process*. Memory and learning are intricately connected. When a particular activity pattern is learned, it is stored in the brain where it can be recalled later when required. Memory may be divided into "short-term" and "long-term" memory, depending on the retention time (Arbib, 1989). *Short-term memory* refers to a compilation of knowledge representing the "current" state of the environment. Any discrepancies between knowledge stored in short-term memory and a "new" state are used to update the short-term memory. *Long-term memory,* on the other hand, refers to knowledge stored for a long time or permanently.

In this section we study an associative memory that offers the following characteristics:

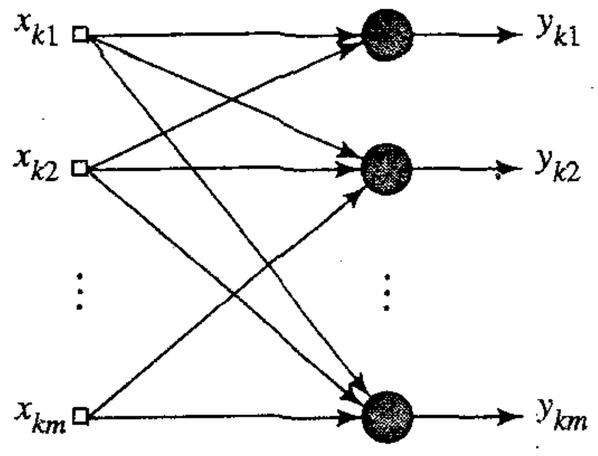
- The memory is distributed.
- Both the stimulus (key) pattern and the response (stored) pattern of an associative memory consist of data vectors.
- Information is stored in memory by setting up a spatial pattern of neural activities across a large number of neurons.
- Information contained in a stimulus not only determines its storage location in memory but also an address for its retrieval.
- Although neurons do not represent reliable and low-noise computing cells, the memory exhibits a high degree of resistance to noise and damage of a diffusive kind.
- There may be interactions between individual patterns stored in memory. (Otherwise the memory would have to be exceptionally large for it to accommodate the storage of a large number of patterns in perfect isolation from each other.) There is therefore the distinct possibility for the memory to make *errors* during the recall process.

In a *distributed memory,* the basic issue of interest is the simultaneous or near-simultaneous activities of many different neurons, which are the result of external or internal stimuli. The neural activities form a spatial pattern inside the memory that contains information about the stimuli. The memory is therefore said to perform a distributed mapping that transforms an activity pattern in the input space into another activity pattern in the output space. We may illustrate some important properties of a



Input layer of neurons | Synaptic junctions | Output layer of neurons

(a) Associative memory model component of a nervous system



Input layer of source nodes | Output layer of neurons

(b) Associative memory model using artificial neurons

FIGURE 2.17 Associative memory models.

distributed memory mapping by considering an idealized neural network that consists of two layers of neurons. Figure 2.17a illustrates a network that may be regarded as a *model component of a nervous system* (Cooper, 1973; Scofield and Cooper, 1985). Each neuron in the input layer is connected to every one of the neurons in the output layer. The actual synaptic connections between the neurons are complex and redundant. In the model of Fig. 2.17a, a single ideal junction is used to represent the integrated effect of all the synaptic contacts between the dendrites of a neuron in the input layer and the axon branches of a neuron in the output layer. The level of activity of a neuron in the input layer may affect the level of activity of every other neuron in the output layer.

The corresponding situation for an artificial neural network is depicted in Fig. 2.17b. Here we have an input layer of source nodes and an output layer of neurons acting as computation nodes. In this case, the synaptic weights of the network are included as integral parts of the neurons in the output layer. The connecting links between the two layers of the network are simply wires.

In the following mathematical analysis, the neural networks in Figs. 2.17a and 2.17b are both assumed to be *linear*. The implication of this assumption is that each neuron acts as a linear combiner, as depicted in the signal-flow graph of Fig. 2.18. To proceed with the analysis, suppose that an activity pattern $\mathbf{x}_k$ occurs in the input layer of the network and that an activity pattern $\mathbf{y}_k$ occurs simultaneously in the output layer. The issue we wish to consider here is that of learning from the association

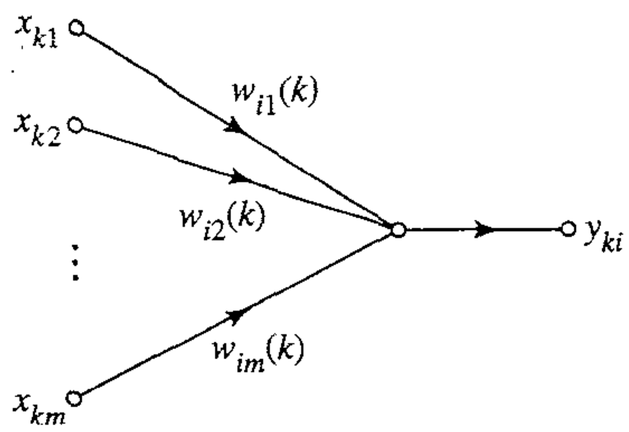


FIGURE 2.18 Signal-flow graph model of a linear neuron labeled *i*.

between the patterns $\mathbf{x}_k$ and $\mathbf{y}_k$. The patterns $\mathbf{x}_k$ and $\mathbf{y}_k$ are represented by vectors, written in their expanded forms as:

$$\mathbf{x}_k = [x_{k1}, x_{k2}, \ldots, x_{km}]^T$$

and

$$\mathbf{y}_k = [y_{k1}, y_{k2}, \ldots, y_{km}]^T$$

For convenience of presentation we have assumed that the input space dimensionality (i.e., the dimension of vector $\mathbf{x}_k$) and the output space dimensionality (i.e., the dimension of vector $\mathbf{y}_k$) are the same, equal to m. From here on we refer to m as *network dimensionality* or simply *dimensionality*. Note that m equals the number of source nodes in the input layer or neurons in the output layer. For a neural network with a large number of neurons, which is typically the case, the dimensionality m can be large.

The elements of both $\mathbf{x}_k$ and $\mathbf{y}_k$ can assume positive and negative values. This is a valid proposition in an artificial neural network. It may also occur in a nervous system by considering the relevant physiological variable to be the difference between an actual activity level (e.g., firing rate of a neuron) and a nonzero spontaneous activity level.

With the networks of Fig. 2.17 assumed to be linear, the association of key vector $\mathbf{x}_k$ with memorized vector $\mathbf{y}_k$ may be described in matrix form as:

$$\mathbf{y}_k = \mathbf{W}(k)\mathbf{x}_k, \qquad k = 1, 2, \ldots, q \tag{2.27}$$

where $\mathbf{W}(k)$ is a weight matrix determined solely by the input–output pair $(\mathbf{x}_k, \mathbf{y}_k)$.

To develop a detailed description of the weight matrix $\mathbf{W}(k)$, consider Fig. 2.18 that shows a detailed arrangement of neuron i in the output layer. The output y_{ki} of neuron i due to the combined action of the elements of the key pattern $\mathbf{x}_k$ applied as stimulus to the input layer is given by

$$y_{ki} = \sum_{j=1}^{m} w_{ij}(k)x_{kj}, \qquad i = 1, 2, \ldots, m \tag{2.28}$$

where the $w_{ij}(k), j = 1, 2, \ldots, m$, are the synaptic weights of neuron i corresponding to the kth pair of associated patterns. Using matrix notation, we may express y_{ki} in the equivalent form

$$y_{ki} = [w_{i1}(k), w_{i2}(k), \ldots, w_{im}(k)]\begin{bmatrix} x_{k1} \\ x_{k2} \\ \vdots \\ x_{km} \end{bmatrix}, \qquad i = 1, 2, \ldots, m \tag{2.29}$$

The column vector on the right-hand side of Eq. (2.29) is recognized as the key vector $\mathbf{x}_k$. By substituting Eq. (2.29) in the definition of the m-by-1 stored vector $\mathbf{y}_k$, we get

$$\begin{bmatrix} y_{k1} \\ y_{k2} \\ \vdots \\ y_{km} \end{bmatrix} = \begin{bmatrix} w_{11}(k) & w_{12}(k) & \dots & w_{1m}(k) \\ w_{21}(k) & w_{22}(k) & \dots & w_{2m}(k) \\ \vdots & \vdots & \vdots & \vdots \\ w_{m1}(k) & w_{m2}(k) & \dots & w_{mm}(k) \end{bmatrix} \begin{bmatrix} x_{k1} \\ x_{k2} \\ \vdots \\ x_{km} \end{bmatrix} \tag{2.30}$$

Equation (2.30) is the expanded form of the matrix transformation or mapping described in Eq. (2.27). In particular, the m-by-m weight matrix $\mathbf{W}(k)$ is defined by

$$\mathbf{W}(k) = \begin{bmatrix} w_{11}(k) & w_{12}(k) & \dots & w_{1m}(k) \\ w_{21}(k) & w_{22}(k) & \dots & w_{2m}(k) \\ \vdots & \vdots & \vdots & \vdots \\ w_{m1}(k) & w_{m2}(k) & \dots & w_{mm}(k) \end{bmatrix} \tag{2.31}$$

The individual presentations of the q pairs of associated patterns $\mathbf{x}_k \rightarrow \mathbf{y}_k$, $k = 1, 2, \dots, q$, produce corresponding values of the individual matrix, namely, $\mathbf{W}(1), \mathbf{W}(2), \dots, \mathbf{W}(q)$. Recognizing that this pattern association is represented by the weight matrix $\mathbf{W}(k)$, we may define an *m–by–m memory matrix* that describes the summation of the weight matrices for the entire set of pattern associations as follows:

$$\mathbf{M} = \sum_{k=1}^{q} \mathbf{W}(k) \tag{2.32}$$

The memory matrix $\mathbf{M}$ defines the overall connectivity between the input and output layers of the associative memory. In effect, it represents the *total experience* gained by the memory as a result of the presentations of q input–output patterns. Stated in another way, the memory matrix $\mathbf{M}$ contains a piece of every input–output pair of activity patterns presented to the memory.

The definition of the memory matrix given in Eq. (2.32) may be restructured in the form of a recursion as shown by

$$\mathbf{M}_k = \mathbf{M}_{k-1} + \mathbf{W}(k), \qquad k = 1, 2, \dots, q \tag{2.33}$$

where the initial value $\mathbf{M}_0$ is zero (i.e., the synaptic weights in the memory are all initially zero), and the final value $\mathbf{M}_q$ is identically equal to $\mathbf{M}$ as defined in Eq. (2.32). According to the recursive formula of Eq. (2.33), the term $\mathbf{M}_{k-1}$ is the old value of the memory matrix resulting from $(k-1)$ pattern associations, and $\mathbf{M}_k$ is the updated value in light of the increment $\mathbf{W}(k)$ produced by the kth association. Note, however, that when $\mathbf{W}(k)$ is added to $\mathbf{M}_{k-1}$, the increment $\mathbf{W}(k)$ loses its distinct identity among the mixture of contributions that form $\mathbf{M}_k$. In spite of the synaptic mixing of different associations, information about the stimuli may not have been lost, as demonstrated in the sequel. Notice also that as the number q of stored patterns increases, the influence of a new pattern on the memory as a whole is progressively reduced.

Correlation Matrix Memory

Suppose that the associative memory of Fig. 2.17b has learned the memory matrix $\mathbf{M}$ through the associations of key and memorized patterns described by $\mathbf{x}_k \rightarrow \mathbf{y}_k$, where $k = 1, 2, \ldots, q$. We may postulate $\hat{\mathbf{M}}$, denoting an *estimate* of the memory matrix $\mathbf{M}$ in terms of these patterns as (Anderson, 1972, 1983; Cooper, 1973):

$$\hat{\mathbf{M}} = \sum_{k=1}^{q} \mathbf{y}_k \mathbf{x}_k^T \tag{2.34}$$

The term $\mathbf{y}_k \mathbf{x}_k^T$ represents the *outer product* of the key pattern $\mathbf{x}_k$ and the memorized pattern $\mathbf{y}_k$. This outer product is an "estimate" of the weight matrix $\mathbf{W}(k)$ that maps the output pattern $\mathbf{y}_k$ onto the input pattern $\mathbf{x}_k$. Since the pattern $\mathbf{x}_k$ and $\mathbf{y}_k$ are both m-by-1 vectors by assumption, it follows that their output product $\mathbf{y}_k \mathbf{x}_k^T$, and therefore the estimate $\hat{\mathbf{M}}$, is an m-by-m matrix. This dimensionality is in perfect agreement with that of the memory matrix $\mathbf{M}$ defined in Eq. (2.32). The format of the summation of the estimate $\hat{\mathbf{M}}$ bears a direct relation to that of the memory matrix defined in that equation.

A typical term of the outer product $\mathbf{y}_k \mathbf{x}_k^T$ is written as $y_{ki} x_{kj}$, where x_{kj} is the output of source node j in the input layer, and y_{ki} is the output of neuron i in the output layer. In the context of synaptic weight $w_{ij}(k)$ for the kth association, source node j acts as a presynaptic node and neuron i in the output layer acts as a postsynaptic node. Hence, the "local" learning process described in Eq. (2.34) may be viewed as a *generalization of Hebb's postulate of learning.* It is also referred to as the *outer product rule* in recognition of the matrix operation used to construct the memory matrix $\hat{\mathbf{M}}$. Correspondingly, an associative memory so designed is called a *correlation matrix memory*. Correlation, in one form or another, is indeed the basis of learning, association, pattern recognition, and memory recall in the human nervous system (Eggermont, 1990.)

Equation (2.34) may be reformulated in the equivalent form

$$\begin{aligned} \hat{\mathbf{M}} &= [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_q] \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_q^T \end{bmatrix} \\ &= \mathbf{Y}\mathbf{X}^T \end{aligned} \tag{2.35}$$

where

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_q] \tag{2.36}$$

and

$$\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_q] \tag{2.37}$$

The matrix $\mathbf{X}$ is an m-by-q matrix composed of the entire set of key patterns used in the learning process; it is called the *key matrix.* The matrix $\mathbf{Y}$ is an m–by–q matrix composed of the corresponding set of memorized patterns; it is called the *memorized matrix.*

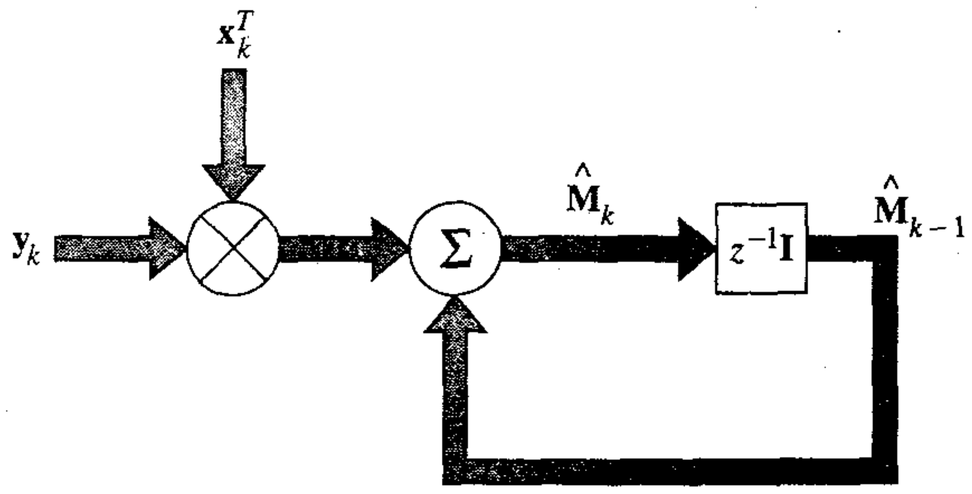


FIGURE 2.19 Signal-flow graph representation of Eq. (2.38).

Equation (2.35) may also be restructured in the form of a recursion as follows:

$$\hat{\mathbf{M}}_k = \hat{\mathbf{M}}_{k-1} + \mathbf{y}_k\mathbf{x}_k^T, \qquad k = 1, 2, \ldots, q \tag{2.38}$$

A signal-flow graph representation of this recursion is depicted in Fig. 2.19. According to this signal-flow graph and the recursive formula of Eq. (2.38), the matrix $\hat{\mathbf{M}}_{k-1}$ represents an old estimate of the memory matrix; and $\hat{\mathbf{M}}_k$ represents its updated value in the light of a new association performed by the memory on the patterns $\mathbf{x}_k$ and $\mathbf{y}_k$. Comparing the recursion of Eq. (2.38) with that of Eq. (2.33), we see that the outer product $\mathbf{y}_k\mathbf{x}_k^T$ represents an estimate of the weight matrix $\mathbf{W}(k)$ corresponding to the kth association of key and memorized patterns, $\mathbf{x}_k$ and $\mathbf{y}_k$.

Recall

The fundamental problem posed by the use of an associative memory is the address and recall of patterns stored in memory. To explain one aspect of this problem, let $\hat{\mathbf{M}}$ denote the memory matrix of an associative memory, which has been completely learned through its exposure to q pattern associations in accordance with Eq. (2.34). Let a key pattern $\mathbf{x}_j$ be picked at random and reapplied as *stimulus* to the memory, yielding the *response*

$$\mathbf{y} = \hat{\mathbf{M}}\mathbf{x}_j \tag{2.39}$$

Substituting Eq. (2.34) in (2.39), we get

$$\begin{aligned} \mathbf{y} &= \sum_{k=1}^{m} \mathbf{y}_k\mathbf{x}_k^T\mathbf{x}_j \\ &= \sum_{k=1}^{m} (\mathbf{x}_k^T\mathbf{x}_j)\mathbf{y}_k \end{aligned} \tag{2.40}$$

where, in the second line, it is recognized that $\mathbf{x}_k^T\mathbf{x}_j$ is a scalar equal to the *inner product* of the key vectors $\mathbf{x}_k$ and $\mathbf{x}_j$. We may rewrite Eq. (2.40) as

$$\mathbf{y} = (\mathbf{x}_j^T\mathbf{x}_j)\mathbf{y}_j + \sum_{\substack{k=1 \\ k \neq j}}^{m} (\mathbf{x}_k^T\mathbf{x}_j)\mathbf{y}_k \tag{2.41}$$

Let each of the key patterns $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_q$ be normalized to have unit energy; that is,

$$\begin{aligned} E_k &= \sum_{l=1}^{m} x_{kl}^2 \\ &= \mathbf{x}_k^T \mathbf{x}_k \\ &= 1, \qquad k = 1, 2, \ldots, q \end{aligned} \tag{2.42}$$

Accordingly, we may simplify the response of the memory to the stimulus (key pattern) $\mathbf{x}_j$ as

$$\mathbf{y} = \mathbf{y}_j + \mathbf{v}_j \tag{2.43}$$

where

$$\mathbf{v}_j = \sum_{\substack{k=1 \\ k \neq j}}^{m} (\mathbf{x}_k^T \mathbf{x}_j) \mathbf{y}_k \tag{2.44}$$

The first term on the right-hand side of Eq. (2.43) represents the "desired" response $\mathbf{y}_j$; it may therefore be viewed as the "signal" component of the actual response $\mathbf{y}$. The second term $\mathbf{v}_j$ is a "noise vector" that arises because of the *crosstalk* between the key vector $\mathbf{x}_j$ and all the other key vectors stored in memory. The noise vector $\mathbf{v}_j$ is responsible for making errors on recall.

In the context of a linear signal space, we may define the *cosine of the angle* between a pair of vectors $\mathbf{x}_j$ and $\mathbf{x}_k$ as the inner product of $\mathbf{x}_j$ and $\mathbf{x}_k$ divided by the product of their individual Euclidean *norms* or *lengths* as shown by

$$\cos(\mathbf{x}_k, \mathbf{x}_j) = \frac{\mathbf{x}_k^T \mathbf{x}_j}{\|\mathbf{x}_k\| \|\mathbf{x}_j\|} \tag{2.45}$$

The symbol $\|\mathbf{x}_k\|$ signifies the Euclidean norm of vector $\mathbf{x}_k$, defined as the square root of the energy of $\mathbf{x}_k$:

$$\begin{aligned} \|\mathbf{x}_k\| &= (\mathbf{x}_k^T \mathbf{x}_k)^{1/2} \\ &= E_k^{1/2} \end{aligned} \tag{2.46}$$

Returning to the situation, note that the key vectors are normalized to have unit energy in accordance with Eq. (2.42). We may therefore reduce the definition of Eq. (2.45) to

$$\cos(\mathbf{x}_k, \mathbf{x}_j) = \mathbf{x}_k^T \mathbf{x}_j \tag{2.47}$$

We may then redefine the noise vector of Eq. (2.44) as

$$\mathbf{v}_j = \sum_{\substack{k=1 \\ k \neq j}}^{m} \cos(\mathbf{x}_k, \mathbf{x}_j) \mathbf{y}_k \tag{2.48}$$

We now see that if the key vectors are *orthogonal* (i.e., perpendicular to each other in a Euclidean sense), then

$$\cos(\mathbf{x}_k, \mathbf{x}_j) = 0, \qquad k \neq j \tag{2.49}$$

and therefore the noise vector $\mathbf{v}_j$ is identically zero. In such a case, the response $\mathbf{y}$ equals $\mathbf{y}_j$. The *memory associates perfectly* if the key vectors from an *orthonormal set;* that is, if they satisfy the following pair of conditions:

$$\mathbf{x}_k^T \mathbf{x}_j = \begin{cases} 1, & k = j \\ 0, & k \neq j \end{cases} \tag{2.50}$$

Suppose now that the key vectors do form an orthonormal set, as prescribed in Eq. (2.50). What is then the limit on the *storage capacity* of the associative memory? Stated in another way, what is the largest number of patterns that can be reliably stored? The answer to this fundamental question lies in the rank of the memory matrix $\hat{\mathbf{M}}$. The *rank* of a matrix is defined as the number of independent columns (rows) of the matrix. That is, if r is the rank of such a rectangular matrix of dimensions l-by-m, we then have $r \leq \min(l, m)$. In the case of a correlation memory, the memory matrix $\hat{\mathbf{M}}$ is an m-by-m matrix, where m is the dimensionality of the input space. Hence the rank of the memory matrix $\mathbf{M}$ is limited by the dimensionality m. We may thus formally state that the number of patterns that can be reliably stored in a correlation matrix memory can never exceed the input space dimensionality.

In real-life situations, we often find that the key patterns presented to an associative memory are neither orthogonal nor highly separated from each other. Consequently, a correlation matrix memory characterized by the memory matrix of Eq. (2.34) may sometimes get confused and make *errors.* That is, the memory occasionally recognizes and associates patterns never seen or associated before. To illustrate this property of an associative memory, consider a set of key patterns.

$$\{\mathbf{x}_{\text{key}}\}: \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_q$$

and a corresponding set of memorized patterns,

$$\{\mathbf{y}_{\text{mem}}\}: \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_q$$

To express the closeness of the key patterns in a linear signal space, we introduce the concept of *community.* We define the community of the set of patterns $\{\mathbf{x}_{\text{key}}\}$ as the lower bound on the inner products $\mathbf{x}_k^T \mathbf{x}_j$ of any two patterns $\mathbf{x}_j$ and $\mathbf{x}_k$ in the set. Let $\hat{\mathbf{M}}$ denote the memory matrix resulting from the training of the associative memory on a set of key patterns represented by $\{\mathbf{x}_{\text{key}}\}$ and a corresponding set of memorized patterns $\{\mathbf{y}_{\text{mem}}\}$ in accordance with Eq. (2.34). The response of the memory, $\mathbf{y}$, to a stimulus $\mathbf{x}_j$ selected from the set $\{\mathbf{x}_{\text{key}}\}$ is given by Eq. (2.39), where it is assumed that each pattern in the set $\{\mathbf{x}_{\text{key}}\}$ is a unit vector (i.e., a vector with unit energy). Let it be further assumed that

$$\mathbf{x}_k^T \mathbf{x}_j \geq \gamma \qquad \text{for } k \neq j \tag{2.51}$$

If the lower bound γ is large enough, the memory may fail to distinguish the response $\mathbf{y}$ from that of any other key pattern contained in the set $\{\mathbf{x}_{\text{key}}\}$. If the key patterns in this set have the form

$$\mathbf{x}_j = \mathbf{x}_0 + \mathbf{v} \tag{2.52}$$

where $\mathbf{v}$ is a stochastic vector, it is likely that the memory will recognize $\mathbf{x}_0$ and associate with it a vector $\mathbf{y}_0$ rather than any of the actual pattern pairs used to train it in the

first place; $\mathbf{x}_0$ and $\mathbf{y}_0$ denote a pair of patterns never seen before. This phenomenon may be termed *animal logic,* which is not logic at all (Cooper, 1973).

2.12 ADAPTATION

In performing a task of interest, we often find that *space* is one fundamental dimension of the learning process; *time* is the other. The *spatiotemporal* nature of learning is exemplified by many of the learning tasks (e.g., control, beamforming) discussed in Section 2.10. Species ranging from insects to humans have an inherent capacity to represent the temporal structure of experience. Such a representation makes it possible for an animal to *adapt* its behavior to the temporal structure of an event in its behavioral space (Gallistel, 1990).

When a neural network operates in a *stationary* environment (i.e., an environment whose statistical characteristics do not change with time), the essential statistics of the environment can, in theory, be *learned* by the network under the supervision of a teacher. In particular, the synaptic weights of the network can be computed by having the network undergo a training session with a set of data that is representative of the environment. Once the training process has completed, the synaptic weights of the network should capture the underlying statistical structure of the environment, which would justify "freezing" their values thereafter. Thus a learning system relies on *memory,* in one form or another, to recall and exploit past experiences.

Frequently, however, the environment of interest is *nonstationary,* which means that the statistical parameters of the information-bearing signals generated by the environment vary with time. In situations of this kind, the traditional methods of supervised learning may prove to be inadequate because the network is not equipped with the necessary means to *track* the statistical variations of the environment in which it operates. To overcome this shortcoming, it is desirable for a neural network to continually *adapt* its free parameters to variations in the incoming signals in a *real-time* fashion. Thus an *adaptive system* responds to every distinct input as a novel one. In other words the learning process encountered in an adaptive system never stops, with learning going on while signal processing is being performed by the system. This form of learning is called *continuous learning* or *learning-on-the-fly*.

Linear adaptive filters, built around a linear combiner (i.e., a single neuron operating in its linear mode), are designed to perform continuous learning. Despite their simple structure (and perhaps because of it), they are widely used in such diverse applications as radar, sonar, communications, seismology, and biomedical signal processing. The theory of linear adaptive filters has reached a highly mature stage of development (Haykin, 1996; Widrow and Stearns, 1985). However, the same cannot be said about nonlinear adaptive filters.[11]

With continuous learning as the property of interest and a neural network as the vehicle for its implementation, the question we need to address is: How can a neural network adapt its behavior to the varying temporal structure of the incoming signals in its behavioral space? One way of addressing this fundamental issue is to recognize that statistical characteristics of a nonstationary process usually change slowly enough for the process to be considered *pseudostationary* over a window of short enough duration. Examples include: