equations governing voltages and conductances and generate individual action potentials. For example, the voltages in the numerical simulations of Figure E-5 were generated by leaky integrate-and-fire model neurons. More about this and other spiking model neurons can be found in works listed in the bibliography at the end of the appendix, as well as in Appendix F.

Suppose that the threshold parameter θ of a McCulloch-Pitts neuron is set at a high value, equal to the total number of inputs. Then the neuron is active if, and only if, all of its synaptic inputs are active. In other words, the output of the neuron is the *conjunction* of its input variables, which is also known as the logical AND operation. Alternatively, the threshold can be set at a low value, equal to one, such that activation of one or more synaptic inputs is enough to activate the neuron. In this case the output of the neuron is the *disjunction* of its input variables, which is also known as the logical OR operation.

Although a McCulloch-Pitts neuron can compute some logical functions, it cannot compute others. A famous example is the exclusive-or (XOR) operation. By definition the XOR operation on two inputs results in “1” if, and only if, exactly one of its inputs is “1.” Thus if both inputs are “1,” the XOR function outputs “0,” while the OR function outputs “1.” Proving that a single McCulloch-Pitts neuron cannot compute the XOR operation is left as an exercise to the reader. However, XOR can be computed by a network of McCulloch-Pitts neurons, as is explained below.

A Network of Neurons Can Compute Any Boolean Logical Function

What functions can be computed by a network of McCulloch-Pitts neurons? Conjunctions and disjunctions are basic building blocks of Boolean logic. The original definition of a McCulloch-Pitts neuron included both inhibitory and excitatory synapses. It turns out that synaptic inhibition can be used for the operation of negation (logical NOT).

Consider a neuron that is spontaneously active and receives a single strong inhibitory synapse. When the inhibitory synapse is inactive, the neuron is spontaneously active. But when the inhibitory synapse is active, the neuron is inactive, silenced by inhibition. In other words, the neuron responds with 1 when its input is 0 but with 0 when its input is 1. This is exactly the NOT operation.

It is well known that any function of Boolean logic can be synthesized by combining the AND, OR, and NOT operations. Because McCulloch-Pitts neurons can compute all of these operations, it follows that networks of McCulloch-Pitts neurons can compute any function of Boolean logic, including XOR.

Why is it important that these models compute Boolean functions? Boolean logic lies at the heart of modern digital computers. The computers on our desktops, and in fact all digital electronic circuits, are designed to implement Boolean logic. When a digital computer runs a software program, it simply executes sequences of logical operations. Thus networks of McCulloch-Pitts neurons can compute the same functions as digital computers.[1]

These facts about networks of McCulloch-Pitts neurons were discovered in the 1940s and 1950s when neural network models played a role in the formal theory of automata and computation. This line of research showed that neural network models have great computational power in principle. Nevertheless, a difficult question remains: How are computations actually performed by brains? This question cannot be answered by formal arguments alone. It is now being addressed both by theoretical and experimental neuroscientists who try to understand how the brain works, and by computer scientists and engineers who create artificial systems that emulate capabilities of the brain.

The notion that a neuron is a device for computing conjunctions and disjunctions is prominent in the ensuing discussion of neural network models of the visual system.

Perceptrons Model Sequential and Parallel Computation in the Visual System

The term *perceptron* was coined in the 1950s by Frank Rosenblatt to describe his neural network models of visual perception. In a perceptron neurons are organized in layers (Figure E-1).[2] The first layer is the input to the network and the last layer the output. Each layer sends synapses only to the next layer, so that information flows in the "forward" direction from the input to the output. Although perceptrons can be constructed from various kinds of model neurons, we will use the simple McCulloch-Pitts neurons.

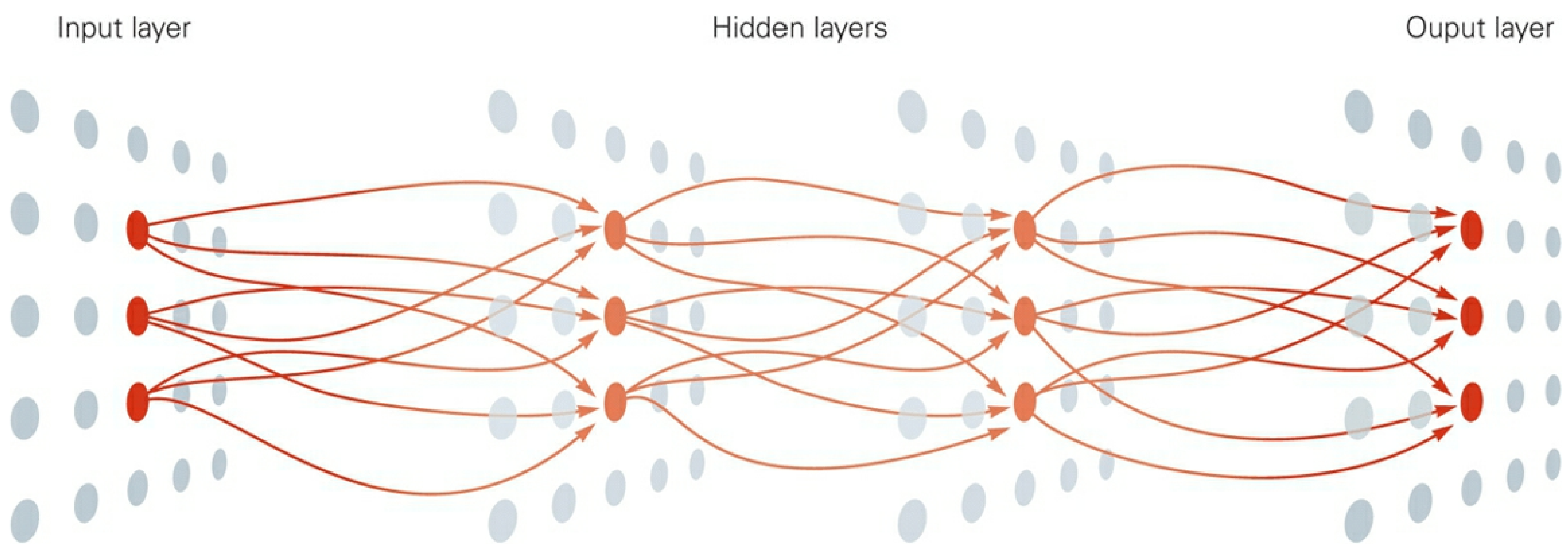


Figure E-1 The perceptron model. A perceptron is a network of idealized neurons arranged in layers with synaptic connections from each layer to the succeeding one. In general, any number of "hidden layers" may intervene between the input and output. Each disk represents a neuron. An arrow pointing from the presynaptic neuron to the postsynaptic neuron represents a synapse. There are no loops in the network.

The computations in a perceptron, as in the visual system, occur through both sequential and parallel processing of information. The layers of a perceptron can be regarded as a sequence of steps in a computation. The neurons within each layer perform similar operations that are executed in parallel during a single step of the computation. Because vision is often quite fast compared to other cognitive tasks, it may require only a few sequential steps, but each step involves a large number of operations performed by many neurons working in parallel. It is natural to represent this kind of computation by a perceptron with a small number of layers, each with many neurons.

Simple and Complex Cells Could Compute Conjunctions and Disjunctions

We shall develop the analogy between perceptrons and the visual system by exploring its implications for primary visual cortex (V1). As discussed in Chapter 27, the "simple cells" of V1 respond selectively to stimuli in the visual field that have a certain spatial orientation. A simple cell responds to a bar of light close to a particular orientation but not to bars with other orientations.

In a classic 1962 paper David Hubel and Torsten Wiesel described this property of orientation selectivity in V1 and also proposed the first model of how it is achieved. They assumed that what they called a "simple" cortical cell receives synaptic inputs from cells in the lateral geniculate nucleus (LGN) and suggested that orientation selectivity of the simple cell in V1 depends on the spatial arrangement of the receptive fields of the LGN cells. Thus, if the center-surround receptive fields of the LGN cells were arranged along a straight line (see Figure 27-3), a bar of light with the same orientation as this line would activate all the LGN inputs of the simple cell simultaneously, driving the cortical simple cell that receives these inputs above the threshold for firing action potentials. Conversely, a bar of light at nonpreferred orientations would stimulate only some of the LGN inputs, leaving that simple cell below threshold for firing.

The preceding model of a simple cell can be interpreted as a McCulloch-Pitts neuron computing an AND operation (Figure E-2A) because a simple cell fires when *all* of its LGN inputs are activated. Recall that a McCulloch-Pitts neuron computes a conjunction if its threshold is set sufficiently high, and intuitively it makes sense that a high threshold goes along with high selectivity.

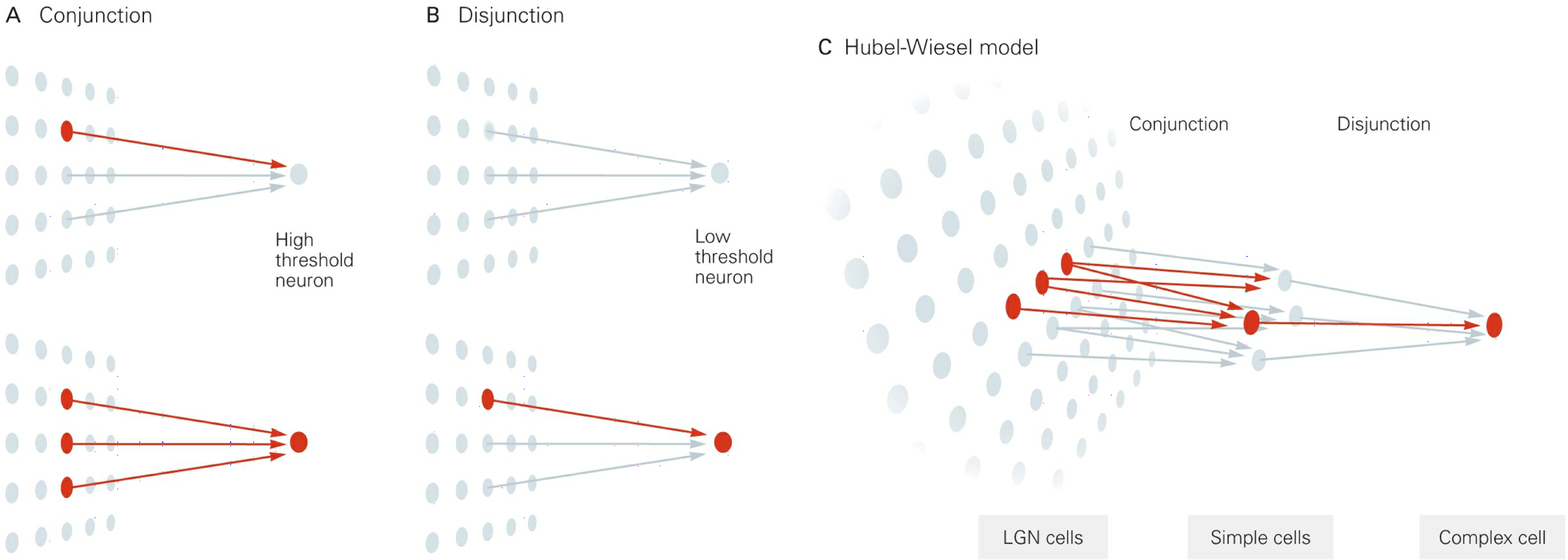


Figure E-2 A perceptron implementing conjunction (AND), disjunction (OR), and the Hubel-Wiesel neurobiological model of simple and complex cells in visual cortex. Neurons are represented by disks and synapses by arrows. Active neurons and synapses are colored red.

A. A neuron with a high threshold can compute the conjunction of three inputs. The neuron does not respond to only one input (top) or two inputs (not shown). It becomes active only when all three inputs are active (bottom).

B. A neuron with a low threshold can compute a disjunction of three inputs. The neuron remains inactive if all of its inputs are inactive (top). It becomes active if a single input neuron is active (bottom) or more than one input neuron is active (not shown).

C. In this realization of the Hubel-Wiesel model a disjunction neuron (right) receives inputs from a set of conjunction neurons (middle), which in turn receive inputs from a grid of neurons (left). The neurons in the grid represent lateral geniculate nucleus (LGN) cells, which are assumed to be either all ON-center or OFF-center cells and retinotopically organized so that the location of each cell in the grid corresponds to the location of its receptive field on the retina. A horizontally oriented visual stimulus activates three LGN cells in a row, which activate a "simple cell" (conjunction) that in turn activates a "complex cell" (disjunction). Like actual simple cells of primary visual cortex, each conjunction neuron responds selectively to stimuli with a particular orientation (horizontal in this case) and at a particular location. Likewise, like actual complex cells, the disjunction neuron responds selectively to stimuli with a particular orientation but is invariant to the exact location of the stimulus.

In addition to simple cells, V1 also contains "complex" cells, also first described by Hubel and Wiesel. Like simple cells, complex cells are orientation selective, but their responses are not sensitive to the location of the stimulus within the receptive field, whereas simple cells are quite sensitive to the precise alignment of the stimulus within the excitatory subregions of their receptive field.

Hubel and Wiesel proposed that a complex cell receives synaptic input from simple cells with similar orientation selectivity (Figure E-2C). The receptive fields of the simple cells add together to form the receptive field of the complex cell. If a visual stimulus with the preferred orientation activates any one of the simple cells, the complex cell is driven over the threshold for firing. This model is intended to explain why spatial location of the stimulus in the receptive field is not a factor in activating the complex cell.

This model of a complex cell can be interpreted as a McCulloch-Pitts neuron computing an OR operation (Figure E-2B) since a complex cell fires when *any* of its simple cell inputs is activated. A McCulloch-Pitts neuron computes a disjunction if its threshold is set sufficiently low, and it makes sense that a low threshold is appropriate for nonselective responses.

In effect, Hubel and Wiesel imagined simple and complex cells as McCulloch-Pitts neurons, although they did not use such language. For a McCulloch-Pitts neuron the threshold determines whether responses are selective or invariant. The simple cell's high threshold is responsible for the cell's orientation selectivity, while the complex cell's low threshold accounts for the invariance of its response to the location of the stimulus within its receptive field.

The Primary Visual Cortex Has Been Modeled As a Multilayer Perceptron

If the Hubel-Wiesel model is extended to many neurons, each with a receptive field that covers a different location in the visual field and tuned to a preferred orientation, then it amounts to a perceptron with three layers of neurons (Figure E-3).

Indeed, like this perceptron, visual areas of the brain generally have a retinotopic organization: Neighboring cells have receptive fields that cover adjacent areas in the visual field. This means that a sheet of cortical tissue functions like a map of the visual field, and patterns of activity can actually resemble images. Similarly, each layer of the model in Figure E-3 is retino-topically organized so that at any moment a map of the overall activity pattern of its neurons depicts the stimulus image. Connections between the layers respect the spatial arrangements of receptive fields described above and shown in Figure E-2. The thresholds are set to yield conjunctions and disjunctions in simple cell and complex cell layers, respectively.

The structure of the model is idealized in a number of ways to facilitate understanding. All cells are arranged in uniformly spaced grids. Furthermore, the simple cell and complex cell layers each have a number of "feature maps." Each cell in a feature map detects exactly the same feature but in a different location of the visual field (Figure E-3). In the cortex the cells detecting different features would be intermingled, but in the model they are segregated for convenience.

A map of active neurons in the LGN layer of the model accurately represents the visual stimulus, whereas the simple and complex cell layers contain more abstract representations of the stimulus because of the orientation selectivity of neurons. In particular, the representation of the stimulus in the complex cell layer is robust and does not reflect small variations in the stimulus (see Figure E-3).

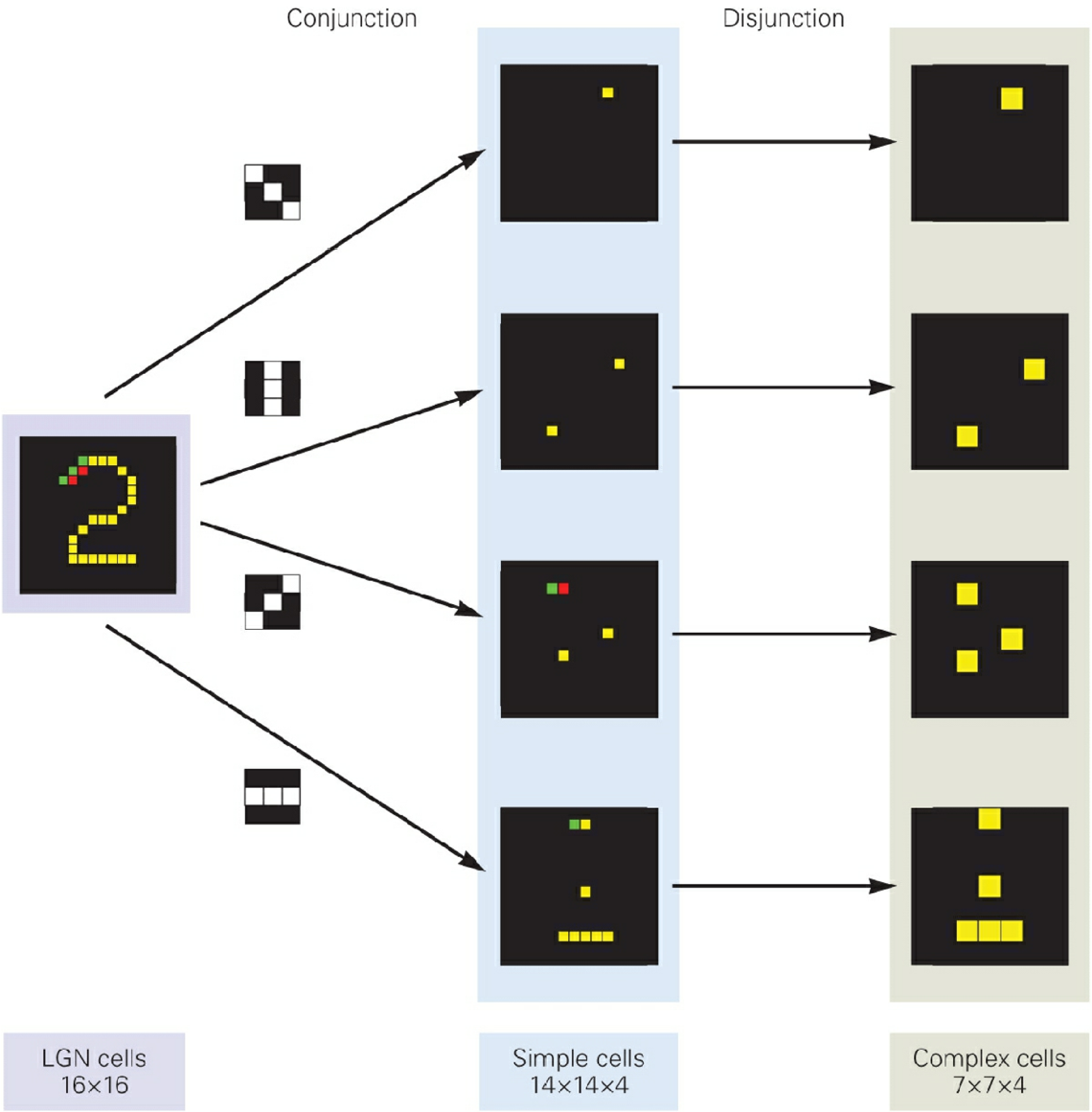


Figure E-3 A perceptron implementing the Hubel-Wiesel model of selectivity and invariance. The network in Figure E-2C can be extended to grids of many cells by specifying synaptic connectivity at all locations in the visual field. The resulting network can be repeated four times, one for each preferred orientation (horizontal, vertical, and two diagonals). This yields four retinotopically organized grids of simple cells, one for each preferred orientation, as well as four grids of complex cells. Each grid is called a *feature map*. Throughout the network the responses to two slightly different images of the numeral 2 are superimposed for comparison. A **yellow pixel** indicates a neuron that responds to both stimuli. A **red pixel** indicates a neuron that responds to one of the stimuli, and a **green pixel** indicates a neuron that responds to the other.

In the LGN layer the difference between the two stimuli is evident (see red and green pixels at the top of the numeral). In the simple cell layer the bottom two feature maps show different responses to the images (red and green pixels), but the top two are the same (all yellow pixels). Finally, the responses of the complex cells are the same for both images (all yellow pixels). Thus invariance and selectivity occur together in one network, although the invariance is limited (it does not hold for all distortions) and the selectivity is fairly simple.

Selectivity and Invariance Must Be Explained by Any Model of Vision

The dichotomy between selectivity and invariance has been important in our discussion of the primary visual cortex and simple stimuli like bars. More generally, this dichotomy is relevant throughout the visual system and even for complex stimuli like entire objects. Let's step back and think about the computations that the entire visual system must accomplish.

Even though the act of seeing appears effortless for humans and animals, vision is a difficult computational problem. In spite of enormous progress in algorithms, speed, and memory capacity, modern digital computers are still far from equaling the performance of biological vision systems. In particular, one of the main functions of vision is the recognition of objects. One reason this task is difficult for computers is that the images of a single object are highly variable. Factors such as lighting, location, and distance all cause changes in retinal images that the visual system must *ignore* in order to recognize an object—recognition requires some invariance in responding. However, the visual system cannot ignore all changes because it has to distinguish between different objects—it must therefore also be selective for certain aspects of images. Although the properties of invariance and selectivity may seem conflicting, they are somehow reconciled by the visual system.

How does the visual system accomplish object recognition? Neurophysiologists have investigated this question by recording from high-level visual areas, such as inferotemporal cortex. To give one example of their findings, certain inferotemporal neurons respond selectively to images of faces. These face-selective neurons have large receptive fields and the exact location of the face within the receptive field is not a factor in the cells' responses. Instead, the responses appear to be closely related to complex features or entire objects rather than simple features like bars or edges.

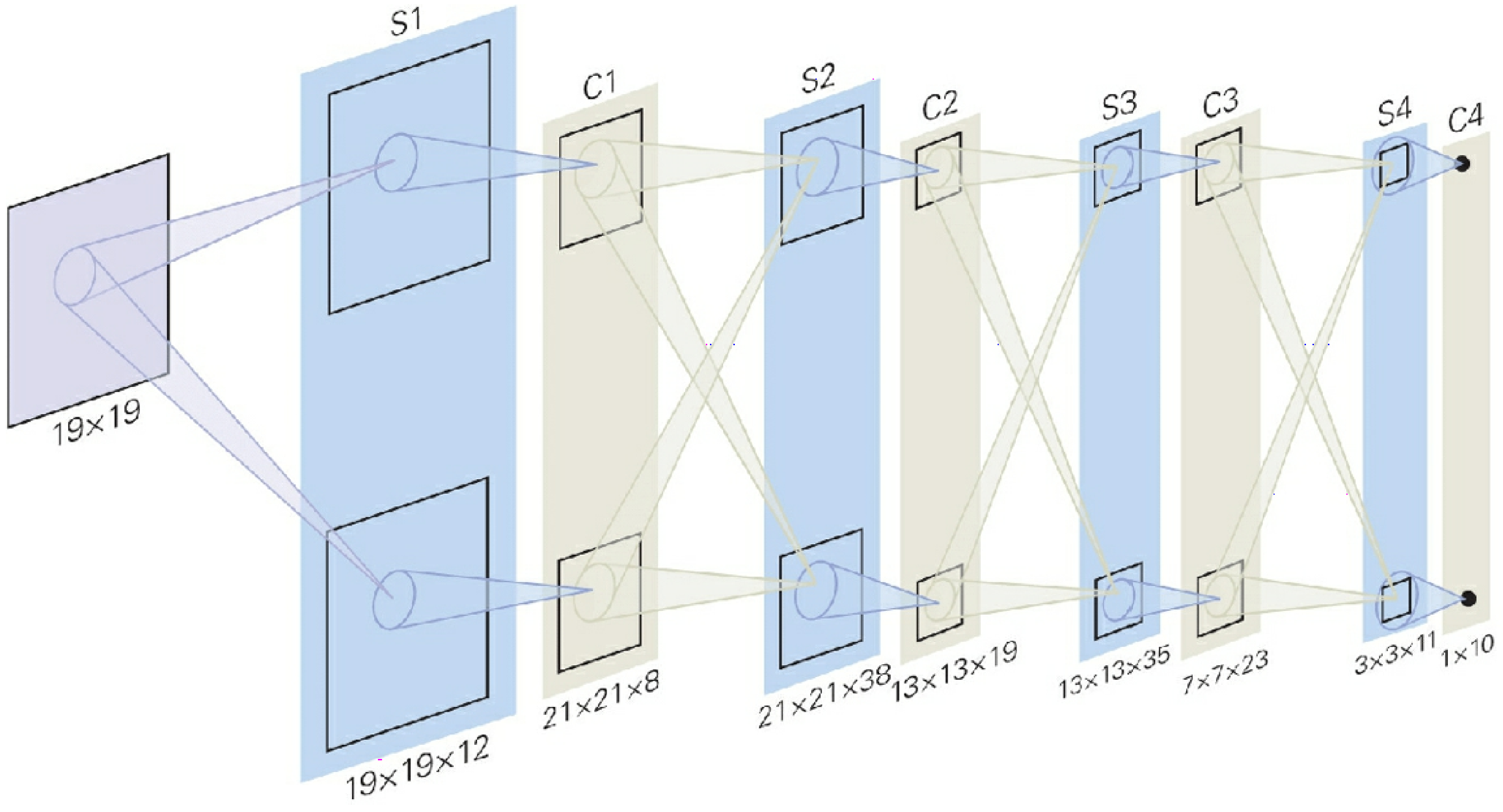
How are selectivity and invariance achieved by the face-selective neurons? According to one theory, all visual areas of cortex are arranged in a hierarchy (see Figure 28-2) and the Hubel-Wiesel model of simple and complex cells in the primary visual cortex (V1) can be generalized to the higher levels of the visual system. In this hierarchical model V1 is at the bottom and areas in the inferotemporal cortex are near the top. Neurons near the bottom of the hierarchy are selective for simple features, have small receptive fields, and are sensitive to small changes in stimulus location. Neurons near the top of the hierarchy are selective for complex features, have large receptive fields, and are invariant to large changes in stimulus location. Neuronal connections from each level to the next are organized so as to carry out computations analogous to the ones performed by simple and complex cells in V1. As we shall see, this hierarchical conception of visual recognition of objects has been formulated precisely in a number of neural network models.

Visual Object Recognition Could Be Accomplished by Iteration of Conjunctions and Disjunctions

Could perceptrons be used to model not just V1 but also the rest of the visual system? We introduced the idea that conjunctions create selectivity in V1 and disjunctions create invariance. Repeated alternation between conjunctions and disjunctions can be used to build up progressively greater selectivity and invariance, culminating in invariant recognition of entire objects.

Indeed, this idea was implemented in 1980 by Kunihiko Fukushima in the *Neocognitron*, a network model designed to recognize handwritten digits. Handwritten numbers may be less complex than images of natural stimuli such as faces or animals, but they are still quite challenging to recognize, as postal workers or anyone who has ever graded handwritten exams can attest. Indeed, digits produced by different writers often look very different, and even repetitions by a single writer can vary considerably.

The Neocognitron has a multilayer, feedforward architecture like that of a perceptron (although inhibition is treated somewhat differently).[3] The first layer functions like a retina in which neurons represent an image of a handwritten digit, and subsequent layers contain multiple feature maps (Figure E-4). Although the first layers are analogous to the layers of simple cells and complex cells of the network in Figure E-3, the subsequent layers are meant to model visual areas of cortex beyond V1. Using Boolean logic as an approximation of the operations performed by the elements in the Neocognitron, one can say that layers alternate between computing conjunctions and disjunctions.[4] In other words, the conjunction-disjunction scheme of the Hubel-Wiesel model is cascaded to form a hierarchical system. In the output layer retinotopic organization disappears completely. There are only 10 output neurons, each of which is selective for one of the digits "0" through "9." In a number of simulations the output neurons show an impressive degree of invariance to the location of the digit in the retina as well as to distortions of the digit.



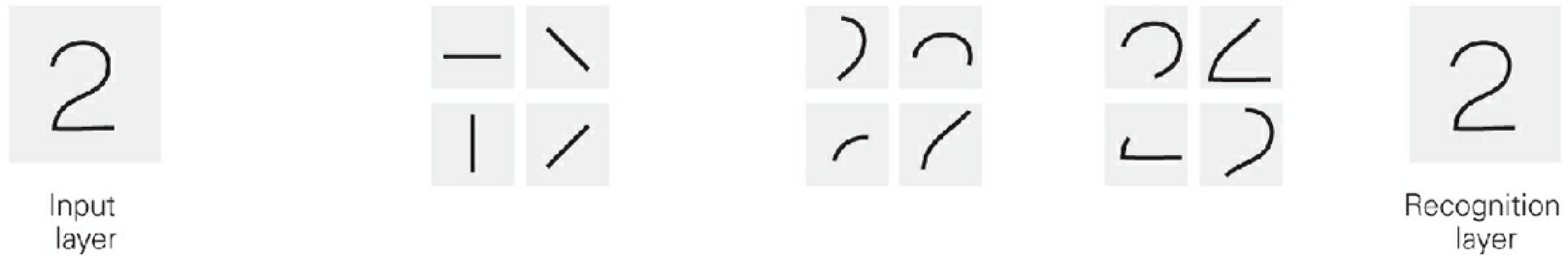


Figure E-4 The Neocognitron model of digit recognition. Each layer in the network is composed of a set of feature maps, and alternating layers contain "S-cells" or "C-cells." All feature maps are retinotopically organized because each cell receives input from neighboring cells of the previous layer. Each cell in a feature map detects the same feature but at different locations in the image.

An S-cell is analogous to a simple cell in the Hubel-Wiesel neurobiological model. It detects conjunction of features detected by C-cells in the previous layer. A C-cell is analogous to a complex cell in the Hubel-Wiesel model. It can be activated by any of the S-cells in the previous layer, which detect the same feature but at slightly different locations in the image. Receptive fields of cells become larger until the retinotopic organization vanishes completely in the final (recognition) layer.

The Neocognitron was constructed for the purpose of recognizing images of handwritten digits. Accordingly, the output neurons are detectors for the digits "0" through "9" and are highly invariant to small variations. Each S-cell layer generates more complex feature selectivity, and each C-cell layer yields more spatial invariance.

The images at the bottom are examples of preferred stimuli of cells in each layer. S1 and C1 cells respond selectively to oriented bars; S2 and C2 cells are selective for more complex features, such as the conjunction of bars; S3 and C3 cells are selective for still more complex features.

A similar model was later developed by Yann LeCun and his colleagues. This model, called *LeNet*, adheres closely to the standard definition of a perceptron. The backpropagation algorithm was used to change the synaptic strengths of LeNet so as to reduce the error rate in recognizing images (Box E-2). LeNet achieved sufficient accuracy in recognizing handwritten characters to be used in some commercial applications. Its descendants are still being used today in the field of computer vision and are competitive with other state-of-the-art approaches.

In the Neocognitron and LeNet the Hubel-Wiesel neurobiological model of V1 is elaborated to the entire process of object recognition. In spite of several decades of intense scrutiny, there remain significant hurdles to testing neural network models of visual processing. To test a model two questions must be addressed. Are there synaptic connections in the brain like those of the model? Is the model a good approximation without other types of connections that are not included? Much experimental evidence concerning these questions is rather indirect and circumstantial. In particular, anatomical techniques for determining the connectivity of cortical circuits are still in their infancy. For example, there is no direct anatomical evidence for the hypothesis that simple cells in V1 are driven by LGN neurons with receptive fields lined up in a row, as originally proposed by Hubel and Wiesel, although there is some indirect physiological evidence.

As mentioned earlier, attempts have been made to arrange visual areas of cortex in a hierarchy that is consistent with the known anatomical connections between areas. When the visual system is modeled as a perceptron, only "bottom-up" connections are included. In reality, however, there are also "top-down" connections. In some cases, such as the pathways between LGN and V1, the top-down connections far outnumber the bottom-up ones. It is thought that top-down connections are important for allowing cognitive factors such as expectation to influence perception.

Box E-2 Learning in Neural Networks

The brain can perform many computational tasks that are beyond the capabilities of today's electronic computers, but it is also remarkable for another reason: It is a self-assembled system, wiring up its own synaptic connections, unlike an electronic computer that is actually built by external agents (humans or machines).

To emulate this process of self-assembly or self-organization, many neural models are equipped with dynamic processes that continually reorganize their synaptic connections. Some processes create or eliminate neurons or their connections, whereas others adjust the strengths of existing synaptic connections or change other properties of neurons.

To describe the process of self-organization, it is helpful to introduce some terminology for describing the synaptic organization of neural networks. The term *synaptic weight* is often used to refer to the strength of a particular synaptic connection, whereas the term *synaptic weight matrix* applies to the set of all synaptic weights in a network. The strength of the synapse onto neuron *i* from neuron *j* is written as W_{ij}. This is the element of the weight matrix located at the intersection of row *i* and column *j* (see Box E-1).

In many neural network models the weight matrix evolves in time according to a *synaptic plasticity rule*, a mathematical model governing the modifications of synaptic strengths. This is often called a *learning rule*, although strictly speaking, learning is a behavior of a network rather than a synapse.

The network typically starts out in a naïve state, that is, the weight matrix is initialized with random values. Then the network is exposed to a series of stimuli, each of which causes the weight matrix to be modified by the learning rule. Learning rules can take many forms. Much effort has been devoted to devising them and exploring their properties. The Hebbian rule is popular in neurobiological models; with this rule synapses are modified based on temporally contiguous activity of presynaptic and postsynaptic neurons.

It is common to apply the same learning rule to all synapses (or sometimes all excitatory synapses). In spite of this uniformity, the weight matrix becomes heterogeneous because the learning rule depends on activity, and activity patterns are typically nonuniform across a network. Therefore, very complex networks can be produced by a simple learning rule.

In some cases the life of the network is separated into training and operating phases. In the training phase synapses change, whereas in the operating phase the learning rules are turned off. This is analogous to natural development in which plasticity seems particularly strong in juvenile animals. In other cases the learning rules may be turned off gradually. In fully online learning the learning rules are never turned off, so that the network is always able to adapt to new situations.

It is commonly assumed that reorganization of neural networks in the brain is a decentralized process in which synapses are modified as a result of the interaction of the pre- and postsynaptic neurons rather than in response to signals from some central authority. The Hebbian rule is an example. A consequence of such localized self-organization is that one synapse on a neuron can be modified while another remains unchanged. Such specificity is generally observed in biological experiments on Hebbian plasticity, although some exceptions have been reported.

In addition to signaling in the pre- to postsynaptic direction, retrograde messengers such as nitric oxide may also play a role in synaptic plasticity (see Chapter 11), although their role has not been extensively explored in models. The diffuse neuromodulatory systems also have effects on synaptic plasticity (see Chapter 13), and some neural network models have attempted to include interaction between global signals from a central source and local signals as a factor in synaptic modification.

Learning rules are sometimes classified as unsupervised or supervised. *Supervised learning* involves an external “teacher” that evaluates the performance of the entire network and sends a reward or error signal that somehow reaches the synapses. The learning rule is devised so that it produces synaptic modifications that improve the performance of the network as evaluated by the teacher.

One of the most popular supervised learning methods is known as *backpropagation*. When implemented in a perceptron an error signal is propagated back through the network, starting with the output neurons and moving toward the input neurons. The synapses are then modified based on neural activity and the backpropagated error signal.

Backpropagation has been used by engineers for practical applications, such as a computer system for recognizing handwritten numbers based on LeNet. However, it is unclear whether backpropagation is a biologically plausible learning mechanism, even if it may be useful for engineers.

Using *Unsupervised learning* rules, such as the Hebbian rule, the network learns from sensory inputs without an explicit error signal. These learning rules can have a number of computational functions, such as associative learning, discovering useful stimulus features, or reducing the dimensionality of complex stimuli. They have been used to model the self-organization of feature maps in the primary visual cortex during the course of neural development (see Box E-3), as well as to train networks like the Neocognitron.

Given these uncertainties and limitations, how useful are perceptrons as models of vision? Although, perceptrons are simplistic—they encompass only a subset of the connections in the visual system—they may capture some essence of the way that neural circuits perform visual computations. Indeed, perceptrons perform impressively on visual tasks such as recognizing handwritten digits, although they still fall short of human performance. Such engineering applications show how far one can push the simple ideas embodied in the perceptron.

Neural networks like the Neocognitron and LeNet model the visual system as a perceptron organized into a hierarchy of feature detectors. These models propose an answer to one of the questions posed at the beginning of this appendix: How is the psychological event of recognizing an object related to the huge number of neural events that underlie it? In a hierarchical perceptron the recognition of an object involves a relatively small number of sequential steps, each of which consists of a large number of operations executed in parallel. Each operation is very simple, carried out by a neuron that is activated when its synaptic inputs drive it above threshold. The sequential steps alternate between selectivity for more complex features and invariance to small distortions of these features. The neurons at the end of this sequence are selective for entire objects, ignoring variations in their appearance. Thus object recognition can be considered as an emergent property of the network, one that requires the coordinated activation of many neurons, located at many different steps.

Fifty years after Rosenblatt's pioneering work it is clear that perceptrons have been important in developing models of computations in the visual system. In the study of visual perception, as in other fields of science, formal models have proved to be valuable aids to experimentalists.

Associative Memory Networks Use Hebbian Plasticity to Store and Recall Neural Activity Patterns

The sight of a familiar face evokes a name. A simple odor triggers the vivid recollection of a past meal and the persons who were there. These everyday experiences illustrate that the facts and ideas stored in our memories are associated with each other. Philosophers and psychologists have argued that association is the basic principle of all mental activity. Neuroanatomists have studied the way that neurons are bound together in a web of synaptic connections. The two traditions converge in an intuitively appealing idea: Perhaps synaptic connections are the material substrate of mental associations.

This idea has been formalized in a number of neural network models of associative memory. A fundamental assumption in these models is that information is transferred back and forth between neural activity and synaptic connections. When novel information first enters the brain it is encoded in a pattern of neural activity. If this information is stored as memory, the neural activity leaves a trace in the brain in the form of modified synaptic connections. The stored information can be recalled when the modified connections again become active. This scheme assumes that synaptic connections remain stable for long periods of time, whereas neural activity is ephemeral and represents immediate experience only.

The transfer of information from neural activity to synapses is hypothesized to occur through Hebbian synaptic plasticity: A long-lasting increase in synaptic efficacy is induced if the presynaptic neuron repeatedly participates in the firing of its postsynaptic neuron (Box E-3). Some prominent forms of long-term potentiation involving the NMDA-type glutamate receptor are regarded as Hebbian (see Chapter 67). Conversely, the transfer of information from synapses to neural activity is thought to occur through a process of pattern completion in which activity spreads through an assembly of neurons coupled by synaptic loops.