2

Neural Networks I:
Subdivision and Hierarchy

Conceptual Outline

2.1 Motivated by the properties of biological neural networks, we introduce simple mathematical models whose properties may be explored and related to aspects of human information processing.

2.2 The attractor network embodies the properties of an associative content-addressable memory. Memories are imprinted and are accessed by presenting the network with part of their content. Properties of the network can be studied using a signal-to-noise analysis and simulations. The capacity of the attractor network for storage of memories is proportional to the number of neurons.

2.3 The feedforward network acts as an input-output system formed out of several layers of neurons. Using prototypes that indicate the desired outputs for a set of possible inputs, the feedforward network is trained by minimizing a cost function which measures the output error. The resulting training algorithm is called back-propagation of error.

2.4 In order to study the overall function of the brain, an understanding of substructure and the interactions between parts of the brain is necessary. Feedforward networks illustrate one way to build a network out of parts. A second model of interacting subnetworks is a subdivided attractor network. A subdivided attractor network stores more than just the imprinted patterns—it stores composite patterns formed out of parts of the imprinted patterns. If these are patterns that an organism might encounter, then this is an advantage. Features of human visual processing, language and motor control illustrate the relevance of composite patterns.

2.5 Analysis and simulations of subdivided attractor networks reveal that partial subdivision can balance a decline in the storage capacity of imprinted patterns with the potential advantages of composite patterns. However, this balance only allows direct control over composite pattern stability when the number of subdivisions is no more than approximately seven, suggesting a connection to the 7 ± 2 rule of short-term memory.

2.6 The limitation in the number of subdivisions in an effective architecture suggests that a hierarchy of functional subdivisions is best for complex pattern-recognition tasks, consistent with the observed hierarchical brain structure.

2.7 More general arguments suggest the necessity of substructure, and applicability of the 7 ± 2 rule, in complex systems.

2.1 Neural Networks: Brain and Mind

The functioning of the brain as part of the nervous system is generally believed to account for the complexity of human (or animal) interaction with its environment. The brain is considered responsible for sensory processing, motor control,language, common sense,logic, creativity, planning, self-awareness and most other aspects of what might be called higher information processing. The elements believed responsible for brain function are the nerve cells—neurons—and the interactions between them. The interactions are mediated by a variety of chemicals transferred through synapses. The brain is also affected by diverse substances (e.g., adrenaline) produced by other parts of the body and transported through the bloodstream. Neurons are cells that should not be described in only one form, as they have diverse forms that vary between different parts of the brain and within particular brain sections (Fig. 2.1.1). Specifying the complete behavior of an individual neuron is a detailed and complex problem. However, it is reasonable to assume that many of the general principles upon which the nervous system is designed may be described through a much-simplified model that takes into account only a few features of each neuron and the interactions between them. This is expected, in part, because of the large number, on the order of 10^{11}, neurons in the brain.

A variety of mathematical models have been described that attempt to capture particular features of the neurons and their interactions. All such models are incomplete. Some models are particularly well suited for theoretical investigations, others for pattern-recognition tasks. Much of the modern effort in the modeling of the nervous system is of commercial nature in seeking to implement pattern-recognition strategies for artificial intelligence tasks.Our approach will be to introduce two of the simpler models of neural networks, one of which has been used for extensive theoretical studies, the other for commercial applications. We will then take advantage of the simple analysis of the former to develop an understanding of subdivision in neural networks. Subdivision and substructure is a key theme that appears in many forms in the study of complex systems.

There have been many efforts to demonstrate the connection between mathematical models of neural networks and the biological brain. These are important in order to bridge the gap between the biological and mathematical models. The additional readings located at the end of this text may be consulted for detailed discussions. We do not review these efforts here; instead we motivate more loosely the arti-

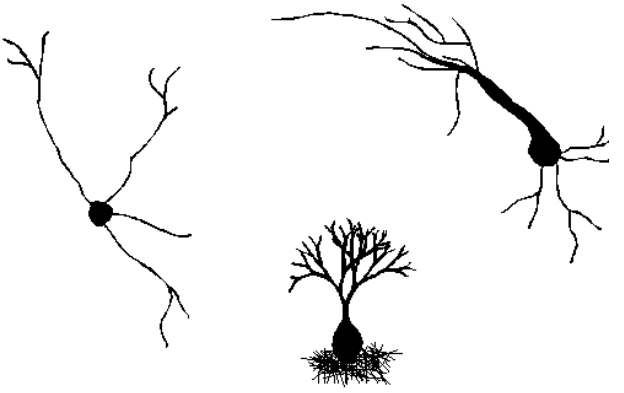

Figure 2.1.1 Several different types of neurons adapted from illustrations obtained by various staining techniques. ▮

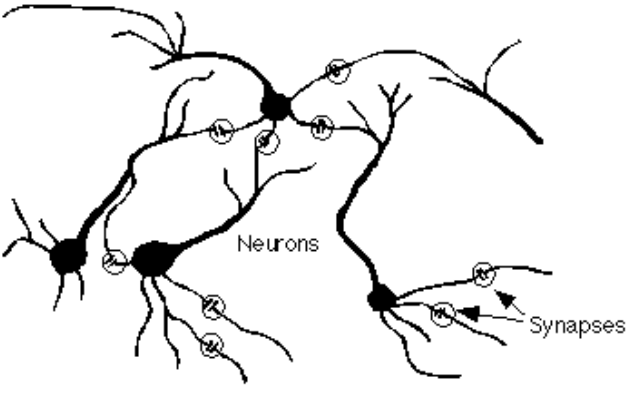

Figure 2.1.2 Schematic illustration of a biological neural network showing several nerve cells with branching axons. The axons end at synapses connecting to the dendrites of the next neuron that lead to its cell body. This schematic illustration is further simplified to obtain the artificial network models shown in Fig. 2.1.3. ▮

ficial models and rely upon investigations of the properties of these models to establish the connection, or to suggest investigations of the biological system.

To motivate the artificial models of neural networks, we show in Fig. 2.1.2 a schematic of a biological neural network that consists of a few neurons.Each neuron has a cell body with multiple projections called dendrites, and a longer projection called an axon which branches into terminal fibers. The terminal fibers of the axon of one neuron generally end proximate to the dendrites of a different cell body. The cell walls of a neuron support the transmission of electrochemical pulses that travel along the axon from the cell body to the terminal fibers. A single electrochemical pulse is not usually considered to be the quantum of information. Instead it is the "activity"—the rate of pulsing—that is considered to be the relevant parameter describing the state of the neuron. Pulses that arrive at the end of a terminal fiber release various chemicals into the narrow intracellular region separating them from the dendrites of the adjacent cell. This region, known as a synapse, provides the medium of influence of one neuron on the next. The chemicals released across the

gap may either stimulate (an excitatory synapse) or depress (an inhibitory synapse) the activity of the next neuron.

It is generally assumed, though not universally accepted, that the "state of the mind" at a particular time is described by the activities of all the neurons—the pattern of neural activity. This activity pattern evolves in time, because the activity of each neuron is determined by the activity of neurons at an earlier time and the excitatory or inhibitory synapses between them. The influence of the external world on the neurons occurs through the activity of sensory neurons that are affected by sensory receptors. Actions are effected by the influence of motor-neuron activity on the muscle cells. Synaptic connections are in part "hardwired" performing functions that are prespecified by genetic programming. However, memory and experience are also believed to be encoded into the strength (or even the existence) of the synapses between neurons. It has been demonstrated that synaptic strengths are affected by the state of neuronal excitation. This influence, called imprinting, is considered to be the principle mechanism for adaptive learning. The most established and well-studied form of imprinting was originally proposed by Hebb in 1949. The plasticity of synapses should not be underestimated, because the development of even basic functions of vision is known to be influenced by sensory stimulation.

Hebbian imprinting suggests that when two neurons are both firing at a particular time, an excitatory synapse between them is strengthened and an inhibitory synapse is weakened. Conversely, when one is firing and the other is not, the inhibitory synapse is strengthened and the excitatory synapse is weakened. Intuitively, this results in the possibility of reconstructing the neural activity pattern from a part of it, because the synapses have been modified so as to reinforce the pattern. Thus, the imprinted pattern of neural activity becomes a memory. This will be demonstrated explicitly and explained more fully in the context of artificial networks that successfully reproduce this process and help explain its function.

The two types of artificial neural networks we will consider are illustrated in Fig. 2.1.3. The first kind is called an attractor network, and consists of mathematical neurons identified as variables s_i that represent the neuron activity. i is the neuron index. Neurons are connected by synapses consisting of variables J_{ij} that represent the strength of the synapse between two neurons i and j. The synapses are taken to be symmetric, so that $J_{ij} = J_{ji}$. A positive value of J_{ij} indicates an excitatory synapse. A negative value indicates an inhibitory synapse. A more precise definition follows in Section 2.2. The second kind of network, discussed in Section 2.3, is called a feedforward network. It consists of a set of two or more layers of mathematical neurons consisting of variables s_i^l that represent the neuron activity. For convenience, l is added as a layer index. Synapses represented by variables J_{ij}^l act only in one direction and sequentially from one layer to the next.

Our knowledge of biological neural networks indicates that it would be more realistic to represent synapses as unidirectional, as the feedforward network does, but to allow neurons to be connected in loops. Some of the effects of feedback in loops are represented in the attractor network by the symmetric synapses.

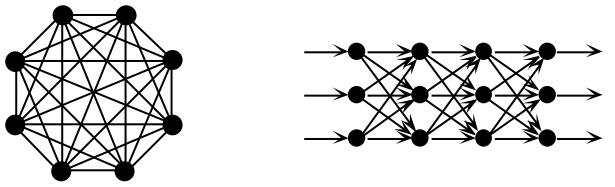

Figure 2.1.3 Schematic illustration of two types of artificial neural networks that are used in modeling biological networks either for formal studies or for application to pattern recognition. On the left is a schematic of an attractor network. The dots represent the neurons and the lines represent the synapses that mediate the influence between them. The synapses are symmetric carrying equal influence in both directions. On the right is a feedforward network consisting of several layers (here four) of neurons that influence each other in a unidirectional fashion. The input arriving from the left sets the values of the first layer of neurons. These neurons influence the second layer of neurons through the synapses between layer one and two. After several stages, the output is read from the final layer of neurons. ∎

A second distinction between the two types of networks is in their choice of representation of the neural activity. The attractor network typically uses binary variables, while the feedforward network uses a real number in a limited range. These choices are related to the nonlinear response of neurons. The activity of a neuron at a particular time is thought to be a sigmoidal function of the influence of other neurons. This means that at moderate levels of excitation, the activity of the neuron is proportional to the excitation. However, for high levels of excitation,the activity saturates. The question arises whether the brain uses the linear regime or generally drives the neurons to saturation. The most reasonable answer is that it depends on the function of the neuron. This is quite analogous to the use of silicon transistors, which are used both for linear response and switching tasks. The neurons that are used in signal-processing functions in the early stages of the auditory or visual systems are likely to make use of the linear regime. However, a linear operation is greatly limited in its possible effects. For example,any number of linear operations are equivalent to a single linear operation. If only the linear regimes of neurons were utilized,the whole operation of the network would be reducible to application of a linear operator to the input information—multiplication by a matrix. Thus, while for initial signal processing the linear regime should play an important role,in other parts of the brain the saturation regime should be expected to be important. The feedforward network uses a model of nonlinear response that includes both linear and saturation regimes, while the attractor network typically represents only the saturation regime.Generalizing the attractor network to include a linear regime adds analytic difficulty, but does not significantly change the results. In contrast, both the linear and nonlinear regimes are necessary for the feedforward network to be a meaningful model.

Each of the two artificial network models represents drastic simplifications over more realistic network models. These simplifications enable intuitive mathematical

treatments and capture behaviors that are likely to be an important part of more realistic models. The attractor network with symmetric synapses is the most convenient for analytic treatments because it can be described using the stochastic field formalism discussed in Section 1.6. The feedforward network is more easily used as an input-output system and has found more use in applications.

2.2 Attractor Networks

2.2.1 *Defining attractor networks*

Attractor networks, also known as Hopfield networks, in their simplest form, have three features:

a. Symmetric synapses:

$$J_{ij} = J_{ji} \tag{2.2.1}$$

b. No self-action by a neuron:

$$J_{ii} = 0 \tag{2.2.2}$$

c. Binary variables for the neuron activity values:

$$s_i = \pm 1 \tag{2.2.3}$$

There are N neurons, so the neuron indices i, j take values in the range $\{1,...,N\}$. By Eq.(2.2.1) and Eq.(2.2.2),the synapses J_{ij} form a symmetric $N \times N$ matrix with all diagonal elements equal to zero.

The binary representation of neuron activity suggests that the activity has only two values which are active or "firing," $s_i = +1$ (ON), and inactive or "quiescent," $s_i = -1$ (OFF). The activity of a particular neuron,updated at time t, is given by:

$$s_i(t) = \text{sign}(\sum_j J_{ij} s_j(t-1)) \tag{2.2.4}$$

where the values of all the other neurons at time $t-1$ are polled through the synapses to determine the ith neuron activity at time t. Specifically, this expression states that a particular neuron fires or does not fire depending on the result of performing a sum of all of the messages it is receiving through synapses. This sum is formed from the activity of every neuron multiplied by the strength of the synapse between the two neurons. Thus, for example,a firing neuron j, $s_j = +1$, which has a positive (excitatory) synapse to the neuron i, $J_{ij} > 0$, will increase the likelihood of neuron i firing. If neuron j is not firing, $s_j = -1$, then the likelihood of neuron i firing is reduced. On the other hand,if the synapse is inhibitory, $J_{ij} < 0$, the opposite occurs — a firing neuron j, $s_j = +1$, will decrease the likelihood of neuron i firing, and a quiescent neuron j, $s_j = -1$, will increase the likelihood of neuron i firing. When necessary, it is understood that sign(0) takes the value ± 1 with equal probability.

The activity of the whole network of neurons may be determined either synchronously (all neurons at once) or asynchronously (selecting one neuron at a time). Asynchronous updating is probably more realistic in models of the brain. However,

for many purposes the difference is not significant, and in such cases we can assume a synchronous update.

2.2.2 *Operating and training attractor networks*

Conceptually, the operation of an attractor network proceeds in the following steps. First a pattern of neural activities, the "input", is imposed on the network. Then the network is evolved by updating several times the neurons according to the neuron update rule, Eq.(2.2.4). The evolution continues until either a steady state is reached or a prespecified number of updates have been performed. Then the state of the network is read as the "output." The next pattern is then imposed on the network.

At the same time as the network is performing this process, the synapses themselves are modified by the state of the neurons according to a mathematical formulation of the Hebbian rule:

$$J_{ij}(t) = J_{ij}(t-1) + cs_i(t-1)s_j(t-1) \qquad i \neq j \tag{2.2.5}$$

where the rate of change of the synapses is controlled by the parameter c. This is a mathematical description of Hebbian imprinting, because the synapse between two neurons is changed in the direction of being excitatory if both neurons are either ON or OFF, and the synapse is changed in the direction of being inhibitory if one neuron is ON and the other is OFF.

The update of a neuron is considered to be a much faster process than the Hebbian changes in the synapses—the synaptic dynamics. Thus we assume that c is small compared to the magnitude of the synapse values, so that each imprint causes only an incremental change. Because the change in synapses occurs much more slowly than the neuron update, for modeling purposes it is convenient to separate it completely from the process of neuron update. We then describe the operation of the network in terms of a training period and an operating period.

The training of the network consists of imprinting a set of selected neuron firing patterns $\{\xi_i^\mu\}$ where i is the neuron index $i \in \{1,...,N\}$, μ is the pattern index $\mu \in \{1,...,p\}$, and ξ_i^μ is the value of a particular neuron s_i in the μth pattern. It is assumed that there are a fixed number p of patterns that are to be trained. The synapses are then set to:

$$J_{ij} = \begin{cases} \dfrac{1}{N}\displaystyle\sum_{\mu=1}^{p} \xi_i^\mu \xi_j^\mu & i \neq j \\ 0 & i = j \end{cases} \tag{2.2.6}$$

The prefactor $1/N$ is a choice of normalization of the synapses that is often convenient, but it does not affect in an essential way any results described here.

2.2.3 *Energy analog*

The formulation of the attractor network can be recognized as a generalization of the Ising model discussed in Section 1.6. Neurons are analogous to spins, and the interaction between two spins s_i and s_j is the synapse J_{ij}.

We can thus identify the effective energy of the system as:

$$E\left[\left\{s_i\right\}\right] = -\frac{1}{2}\sum_{i,j} J_{ij}s_i s_j \tag{2.2.7}$$

The update of a particular neuron, Eq.(2.2.4), consists of "aligning" it with the effective local field (known as the postsynaptic potential):

$$h_i(t) = \sum_{j\neq i} J_{ij}s_j(t-1) \tag{2.2.8}$$

This is the same dynamics as the Glauber or Monte Carlo dynamics of an Ising model at zero temperature. At zero temperature the system evolves to a local minimum energy state. In this state each spin is aligned with the effective local field.

The analogy between a neural network and a model with a well-defined energy enables us to consider the operation of the network in a natural way. The pattern of neural activities evolves in time to decrease the energy of the pattern until it reaches a local energy minimum, where each neuron activity is consistent with the influences upon it as measured by the postsynaptic potential. Imprinting a pattern of neural activity lowers the energy of this pattern and, to a lesser degree,the energy of patterns that are similar. In lowering the energy of these patterns,imprinting creates a basin of attraction. The basin of attraction is the region of patterns near the imprinted pattern that will evolve under the neural updating back to the imprinted pattern (Fig. 2.2.1).

The network operation can now be understood schematically as follows (Fig. 2.2.2). We imprint a particular pattern. If we then impose a different pattern on the network, the evolution of the neurons will recover the original pattern if the imposed pattern is within its basin of attraction. The more similar are the two patterns, the more likely the imposed pattern will be in the basin of attraction of the imprinted one. Since part of the imprinted pattern was retrieved, the network acts as a kind of memory.

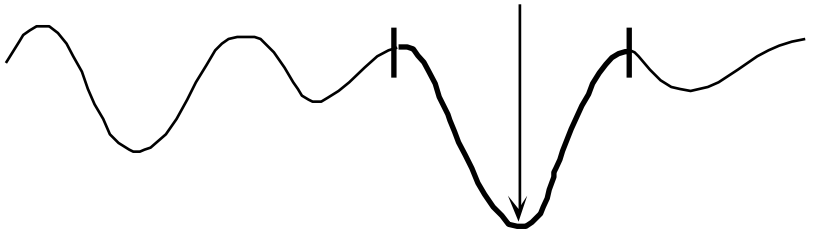

Figure 2.2.1 Schematic illustration of the energy analog of imprinting on an attractor network. Imprinting a pattern lowers its energy and the energy of all patterns in its vicinity. This creates a basin of attraction. If we initialize the network to any pattern within the basin of attraction, the network will relax to the imprinted pattern by its own neural evolution. The network acts as a memory that is "content-addressable." When a pattern is imprinted we can recover it by starting from partial information about it (see Fig. 2.2.2). This is also a form of associative memory. ∎

The operation of the network may be described as an associative memory. The restoration of the complete imprinted pattern, in effect, associates the reconstructed part of the pattern with the part of the pattern that was imposed. We can also say that the network has the ability to perform a kind of generalization (Fig. 2.2.3). The network has generalized the imprinted pattern to the set of patterns that are in its basin of attraction. Moreover, the retrieval process is also a form of categorization,since it assigns the imprinted pattern as a category label to the set of patterns in the basin of attraction. All of these properties of the neural network are suggestive of some of the basic features that are thought to apply to human memory, and thus to the biological neural network. Their natural relationship to each other and the simplicity by which they are achieved in this model is one of the main reasons for the interest in this neural network representation.

We can contrast the properties of the network memory with a computer memory. In a computer, the memory is accessed by an address that specifies the location of a particular piece of information. In order to retrieve information, it is necessary to have the address, or to search systematically through the possibilities. On the other hand, for a human being, retrieving the rest of the sentence "To be or not to be …"is generally much easier than retrieving line 64 from act 3, scene 1, of *Hamlet*, by William Shakespeare. To emphasize the difference in the nature of addressing between

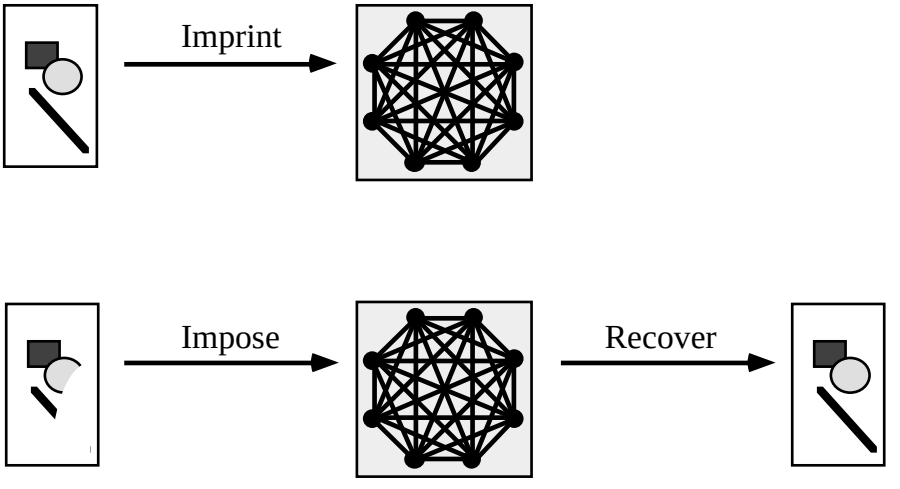

Figure 2.2.2 Schematic operation of the attractor network as a content-addressable memory. Imprinting a pattern on the network in the training stage (top) enables us to use the network as a content-addressable memory (bottom). By imposing a pattern that has a significant overlap with the imprinted pattern the original pattern can be recovered. This is analogous to being able to complete the sentence "To be or not to be …" ∎

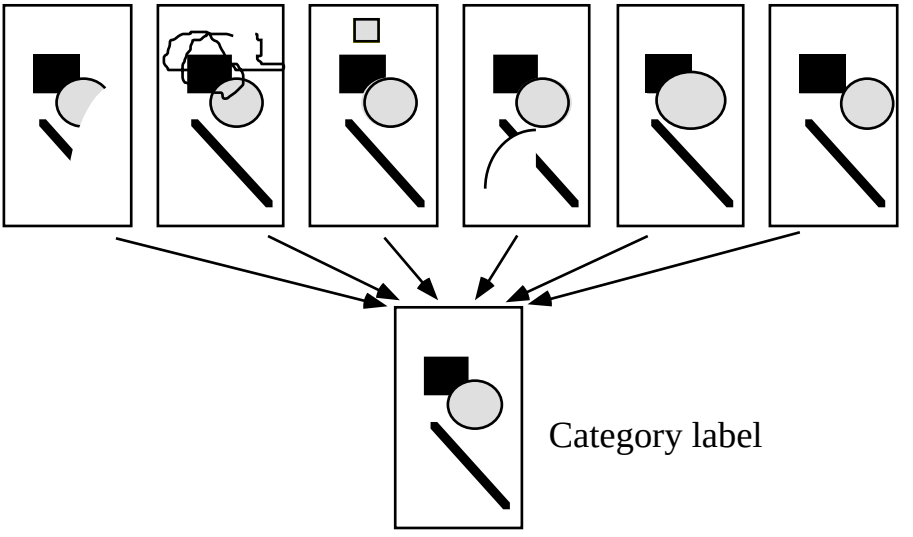

Figure 2.2.3 The neural dynamics of an attractor network maps a variety of patterns onto an imprinted pattern. This is equivalent to a classification of patterns by a category label, which is the imprinted pattern. The category of patterns labeled by the imprinted pattern is its basin of attraction. Classification is also a form of pattern recognition. Moreover, we can say that the basin of attraction is a generalization of the imprinted pattern. Thus the attractor network has properties very unlike those of a conventional computer memory, which is accessed using a numerical address that is distinct from the memory itself. It works much more like human memories that are accessed through information related to the information that is sought after.

The behavior of the attractor network may be summarized as follows:

Attractor network training and operation:
Training — Imprint a neural state.
Operation — Recover original state from part of it.

Analogies for operation:
- Content-addressable memory
- Limited form of classification
- Limited form of pattern recognition
- Limited form of generalization

The relationship between human information-processing and various network models will be discussed further in Chapter 3. ▮

a computer memory and a network memory, we say that the network memory is content addressable.

The associative nature of the attractor network thus captures some of the properties of human memory that are quite distinct from those of a computer memory. It

should be understood, however, that this is not expected to be the last word in development of such models or in the understanding of these processes.

One of the important properties of a memory is its capacity. If we try to imprint more than one pattern on the network, the basin of attraction of each pattern will take up some of the space of all possible patterns. It is natural to expect that there will be a limit to how many patterns we can imprint before the basins of attraction will interfere destructively with each other. When the memory is full, the basins of attraction are small and the memory is not usable because it can only recover a pattern if we already know it. When the destructive interference is complete, the basins of attraction disappear. At that point, a typical imprinted pattern will no longer even be stable, because stability is a basin of attraction of one. Studying the number of imprints that are possible before the network reaches this condition gives us an understanding of the network capacity and how this capacity depends on network size. Thus we can determine the storage capacity by measuring the stability of patterns that are imprinted on the network.

Our mathematical study of attractor networks begins in Section 2.2.4 with an analysis of the network behavior when there are a few imprints. This analysis shows the retrieval of patterns and the relevance of their basin of attraction. In Section 2.2.5 we use a signal-to-noise analysis to determine the stability of an imprinted pattern, and thus the storage capacity of the network. Simulations of an attractor network are discussed in Section 2.2.6. Finally, some aspects of the overload catastrophe that occurs when network capacity is exceeded are discussed in Section 2.2.7.

2.2.4 *One or two imprinted patterns*

We first consider the case of imprinting a single pattern $\{\xi_i\}$. The synapses are constructed as the "outer product" of the neural activities and we have:

$$J_{ij} = \begin{cases} \frac{1}{N}\xi_i\xi_j & i \neq j \\ 0 & i = j \end{cases} \tag{2.2.9}$$

Using these synapses, we start the network at a set of neural activities $\{s_i(0)\}$ and evolve the network using the definition of the dynamics (Eq. (2.2.4)):

$$\begin{aligned} s_i(1) &= \mathrm{sign}\Big(\sum_j J_{ij}s_j(0)\Big) = \mathrm{sign}\Big(\sum_{j\neq i}\frac{1}{N}\xi_i\xi_j s_j(0)\Big) = \mathrm{sign}(\xi_i)\,\mathrm{sign}\Big(\sum_{j\neq i}\xi_j s_j(0)\Big) \\ &= \xi_i\,\mathrm{sign}\Big(\sum_{j\neq i}\xi_j s_j(0)\Big) \end{aligned} \tag{2.2.10}$$

The second line is a consequence of the normalization $|\xi_i| = |\pm 1| = 1$.

If we didn't have the restriction of $j \neq i$ in the sum in Eq.(2.2.10), the factor multiplying ξ_i would be independent of i. We would then have

$$s_i(1) = \xi_i\,\mathrm{sign}\Big(\sum_j \xi_j s_j(0)\Big) = \pm\xi_i \tag{2.2.11}$$

where the ± sign in front is independent of i. This means that in one iteration the network neurons reached either the imprinted pattern or its inverse. This implies that recall of the pattern has been achieved. Why either the pattern or its inverse? It shouldn't be too surprising that we can arrive at either the imprinted pattern or its inverse because the form of the energy function (Eq.(2.2.7)) and the neural update (Eq.(2.2.4)) is invariant under the transformation $s_i \rightarrow -s_i$ for all i simultaneously. Thus,in an attractor network we automatically store both the pattern and its inverse.

How do we treat the actual case with the $i = j$ term missing? We write

$$s_i(1) = \xi_i \,\mathrm{sign}\Big(\sum_j \xi_j s_j(0) - \xi_i s_i(0)\Big) = \xi_i \,\mathrm{sign}(\xi\ \mathbf{s}(0) - \xi_i s_i(0)) \tag{2.2.12}$$

where $\xi\ \mathbf{s}(0) = \sum \xi_j s_j(0)$ is the inner product of the imprinted pattern with the initial state of the network. As long as this inner product is greater than 1 or less than –1, the extra term doesn't affect the result. This means that for $|\xi\ \mathbf{s}(0)| > 1$, recall is achieved and $\mathbf{s}(0)$ is within the basin of attraction of the imprinted pattern or its inverse. This is nearly all of the possible choices for $\mathbf{s}(0)$. Note that the imprinted pattern is a stable fixed point of the network dynamics—once the imprinted pattern is reached it will be repeated.

The case of a single imprinted pattern is somewhat unusual in that even if the initial pattern $\{s_i(0)\}$ is not correlated with the imprinted pattern, we will still recover the imprinted pattern. If we take random numbers for $\{s_i(0)\}$, then the sum over j in Eq. (2.2.12) is a random walk—the sum over N uncorrelated values of ±1 (Section 1.2). The typical size of this number is $\sqrt{N}$, which places the pattern solidly within the basin of attraction of the imprinted pattern or its inverse. It has been suggested that the case of a single dominant imprinted pattern has properties analogous to human obsession, compulsion or fixation—because the imprinted pattern is the output regardless of the input—and is a natural mode of failure of the network that can arise in Hebbian imprinting.

We can also ask what will happen if the magnitude of $\xi\ \mathbf{s}(0)$ is equal to $-1, 0$, or 1. In these cases the first iteration should not lead to the imprinted pattern. However, it is still most likely that after two updates the network will be in either the imprinted pattern or its inverse. When the inner product $\xi\ \mathbf{s}(0)$ is +1, the result of the first update is a new pattern $\mathbf{s}(1)$ that is likely to have a larger than unit overlap with ξ, $\xi\ \mathbf{s}(1) > 1$. When $\xi\ \mathbf{s}(0)$ is -1, it is likely that $\xi\ \mathbf{s}(1) < -1$. The second iteration would be analogous to Eq.(2.2.12):

$$\begin{aligned} s_i(2) &= \xi_i \,\mathrm{sign}\Big(\sum_j \xi_j s_j(1) - \xi_i s_i(1)\Big) \\ &= \xi_i \,\mathrm{sign}(\xi\ \mathbf{s}(1) - \xi_i s_i(1)) \end{aligned} \tag{2.2.13}$$

resulting in retrieval of the imprinted pattern.

The case of $\xi\ \mathbf{s}(0) = 0$ is special, and a synchronous update in this case simply leads to oscillation of the pattern—a 2-cycle:

$$\begin{aligned} s_i(1) &= \xi_i \,\mathrm{sign}\Big(\sum_j \xi_j s_j(0) - \xi_i s_i(0)\Big) \\ &= \xi_i \,\mathrm{sign}(-\xi_i s_i(0)) = -\xi_i^2 s_i(0) = -s_i(0) \end{aligned} \tag{2.2.14}$$

More generally:

$$s_i(t+1) = -s_i(t) \tag{2.2.15}$$

This is one of the few cases where asynchronous updating would lead to a different result,since the randomness inherent in the asynchronous updating would lead to the evolution of the network to the imprinted pattern or its inverse.

We ran into some additional trouble in the preceding discussion because of the omission of the $i = j$ term in the synapses. Why don't we just include this term? The answer, from the point of view of the formal analysis, is that this corresponds to self-action by a neuron on itself. Such self-action is inconsistent with an energy function. In a real network, self-action is not impossible. It might correspond, for example, to an inhibition of neural activity during a period of time after activity has happened. This does occur in biological neurons where the period of self-inhibition is known as a refractory period. The implications of such terms are,however, outside the present discussion.

Our conclusion from the analysis of the single imprint case is that the basin of attraction of the single imprint is large. To measure the size of the basin of attraction, we define the Hamming distance $d(\mathbf{s},\mathbf{s}')$ between two patterns as the number of neurons that differ between the two patterns. The Hamming distance is related to the inner product by

$$d(\mathbf{s},\mathbf{s}') = \frac{N}{2} - \frac{1}{2}\sum_i s_i s_i' \tag{2.2.16}$$

as can be verified using a few examples. The Hamming distance of a pattern from itself is zero, from an orthogonal state is $N/2$, and from its opposite is N. For a single imprint in a neural network the initial pattern $\mathbf{s}(0)$ is in the basin of attraction of the imprinted pattern ifthe inner product between them is positive. This implies that the Hamming distance must be less than $N/2$. This is the effective size (radius) of the basin of attraction.

We can now ask what happens if two patterns are imprinted instead of just one. The synapses are given by Eq.(2.2.6) with $p = 2$. Following through the same steps we have the expression:

$$s_i(1) = \text{sign}\Big(\sum_j J_{ij}s_j(0)\Big) = \text{sign}\Big(\xi_i^1\sum_{j\neq i}\xi_j^1 s_j(0) + \xi_i^2\sum_{j\neq i}\xi_j^2 s_j(0)\Big) \tag{2.2.17}$$

Qualitatively, we can understand this result by considering an initial pattern $\mathbf{s}(0)$ that is close to one of the imprinted patterns,say ξ^1. Let us assume that there is no particular relationship between the two patterns that were imprinted and, quite reasonably, that there is also no relationship between the initial pattern $\mathbf{s}(0)$ and the second imprinted pattern ξ^2. The first sum in Eq. (2.2.17) will give us a number that has a magnitude $N - 2d(\mathbf{s},\xi^1)$. The assumption that the initial configuration is close to the first imprinted pattern means that $d(\mathbf{s},\xi^1)$ is small. The magnitude of the second sum is given by the inner product of the initial pattern with the second imprinted pattern. If there is no relationship between them,then each term in the inner product is independent.

Since each term has the value ±1 with equal probability, it is a random walk with $(N-1)$ steps. The typical magnitude of the second term is thus $\sqrt{N-1}$. For large enough N there is essentially no chance that the second term is as large as the first term. Neglecting the second term, the first term gives us the same result we had before, which is recovery of the imprinted pattern. We see from this argument that retrieval depends on the proximity of the initial state with the pattern that will be retrieved. If the initial pattern is close to the second imprinted pattern,then the second imprinted pattern will be retrieved. Successful retrieval also depends on the number of neurons in the network.

The retrieval of two patterns can be extended to more patterns. For a large enough number of neurons, retrieval will still occur. We can make this argument more rigorous by considering a"signal-to-noise" analysis that pits the term that is trying to retrieve the pattern—the signal—against the rest of the terms—the noise. To do this formally we will assume that all the imprinted patterns are truly uncorrelated with each other. The neural activities are randomly selected values ±1. These values are fixed over the duration of the discussion. In the language of Section 1.3 they are quenched variables. Including correlations between the patterns would be important in understanding how the real brain works. We will consider correlations between patterns later in this chapter in Section 2.4.

2.2.5 *Signal-to-noise analysis of memory stability*

In this section, we formulate what is called a signal-to-noise analysis that enables us to determine statistically the stability of an imprinted pattern. This in turn enables us to determine the storage capacity of the network. Question 2.2.1 generalizes the analysis to give an estimate of the basin of attraction of an imprinted pattern.

We start from a network imprinted with p uncorrelated patterns. From Eq. (2.2.4) and Eq. (2.2.8), an imprinted pattern,imposed as the neural state $\{s_i|s_i = \xi_i^\mu\}$, is stable when $s_i = \text{sign}(h_i)$ or equivalently:

$$s_i h_i > 0 \qquad i \quad \{1,...,N\} \tag{2.2.18}$$

which implies that the local field at each neuron has the same sign as the value of the imprinted pattern.

Without loss of generality, we consider the stability (retrieval) of the first pattern $\{s_i|s_i = \xi_i^1\}$, since by symmetry the choice of pattern is arbitrary. To simplify slightly the notation, we consider the stability of the first neuron s_1, from the results we will be able to infer the stability of the others. The stability of s_1 depends on the sign of

$$s_1 h_1 = s_1 \sum_{j=2}^{N} J_{1j} s_j = \xi_1^1 \sum_{j=2}^{N} J_{1j} \xi_j^1 \tag{2.2.19}$$

Inserting the Hebbian form for the synapses after p imprints, Eq. (2.2.6), we have:

$$s_1 h_1 = \frac{1}{N} \sum_{j=2}^{N} \sum_{\mu=1}^{p} \xi_1^1 \xi_1^\mu \xi_j^\mu \xi_j^1 \tag{2.2.20}$$

We separate this expression into two parts. The first part is due to the imprint of the pattern we are trying to retrieve. The other part is due to all of the other patterns:

$$\begin{aligned} s_1 h_1 &= \frac{1}{N}\sum_{j=2}^{N} \xi_1^1 \xi_1^1 \xi_j^1 \xi_j^1 + \frac{1}{N}\sum_{j=2}^{N}\sum_{\mu=2}^{p} \xi_1^1 \xi_1^\mu \xi_j^\mu \xi_j^1 \\ &= \frac{N-1}{N} + \frac{1}{N}\sum_{j=2}^{N}\sum_{\mu=2}^{p} \xi_1^1 \xi_1^\mu \xi_j^\mu \xi_j^1 \end{aligned} \tag{2.2.21}$$

The first sum was explicitly evaluated because the square of either ±1 is 1. The second sum we can also evaluate, at least statistically. Since ξ_1^μ is not correlated with ξ_j^μ for $j \neq 1$, and ξ_j^1 is not correlated with ξ_j^μ for $\mu \neq 1$, the four factors in each term of the sum are independent of each other. Moreover, each term in the sum has a factor that is independent of the factors in every other term. Therefore, each term in this sum is ±1 with equal probability, and each term is uncorrelated with the others. Thus it is just a random walk with $(N-1)(p-1)$ terms.

We can see that the two parts of $s_1 h_1$ play distinct roles. The first part, called the signal, is positive, and therefore tries to satisfy the stability condition of the imprinted pattern. This is consistent with the idea that a Hebbian imprint contributes to stability of the imprinted pattern. For $N >> 1$, the size of the signal is 1.

The second part, called the noise, can be either positive or negative. The average value of the noise is zero, but the typical value (root mean square value) is given by:

$$\sigma = \frac{1}{N}\sqrt{(N-1)(p-1)} \approx \sqrt{\frac{p}{N}} \tag{2.2.22}$$

where the latter expression is valid for $N, p >> 1$. When the typical value of the noise is much less than the size of the signal, then most of the time the neurons will be stable and we can say that we have a stable imprinted pattern. When the noise is the same size as the signal, then each neuron may either stay the same or switch after a single update, and the pattern will not be stable.

From the expression for the noise term, we see that the stability of the pattern depends on p, the number of patterns that are imprinted on the network. For low storage, $p << N$, the noise term becomes negligible and the imprinted patterns are stable.

If we want to understand how large p can be before the storage will deteriorate, we need to estimate the probability that a single neuron will be unstable—the probability that $s_i h_i$ is negative (see Fig. 2.2.4). The probability that a particular neuron will be unstable is given by the probability that the noise is less than −1. This depends on the distribution of the values of the noise, not just its typical value. We can find the distribution of the noise using the central limit theorem (Section 1.2). When the number of steps in the random walk is large, the distribution of values of the noise can be approximated by a Gaussian (Eq. (1.2.39)). Then we can find the probability that a neuron is unstable using:

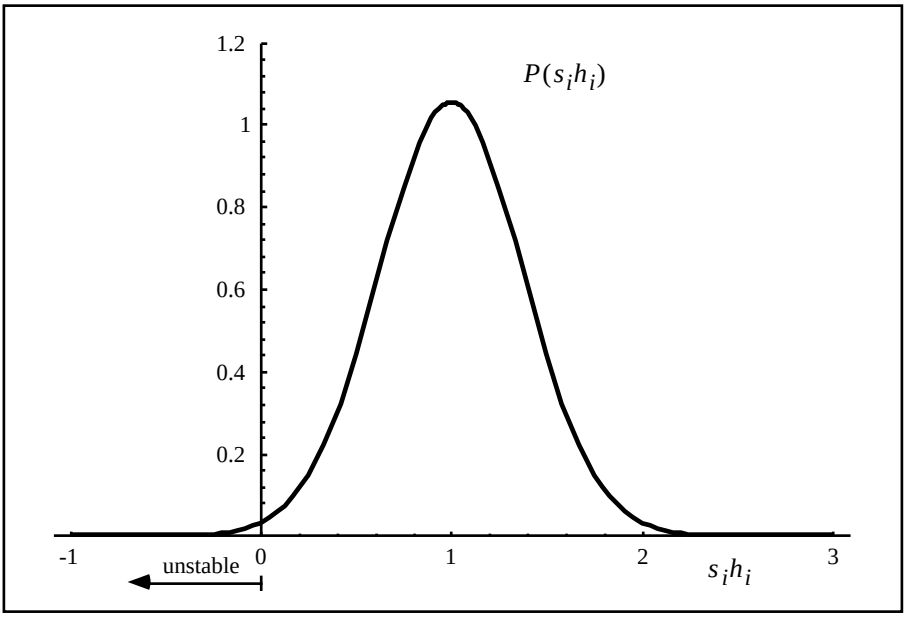

Figure 2.2.4 The probability distribution of the neuron activity times the local field $s_i h_i$. This figure illustrates the signal-to-noise analysis of the stability of an imprinted pattern. The average value of $s_i h_i$ (the signal) is 1. The standard deviation σ of the distribution $P(s_i h_i)$ (the noise) is given by Eq. (2.2.22). Neurons that are unstable have a negative value of $s_i h_i$. The figure is drawn for $\sigma = .379$, when less than 1% are unstable. If σ is larger than this critical value there are more unstable neurons, and when they switch after one update of the network they destabilize the whole pattern. When σ is smaller than this critical value and there are fewer unstable neurons, the rest of the pattern remains stable. The critical value of σ corresponds to a maximum number of patterns that can be stored in the network. ∎

$$P(s_i h_i < 0) = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{-1} dx\, e^{-x^2/2\sigma^2} = \frac{1}{2}\left[1 - \mathrm{erf}\left(\frac{1}{\sigma\sqrt{2}}\right)\right] \tag{2.2.23}$$

The latter expression follows from the definition of the error function erf(x). For $\sigma < 0.430$ this probability, and therefore the fraction of unstable neurons in the imprinted pattern, is less than 1%. The unstable neurons will switch their values when the network updates itself. We now make the assumption that a few unstable neurons will not, when they flip their values, destabilize many other neurons. This makes sense for few enough unstable neurons. If we are satisfied with a small fraction of error of about 1%, we can store a number of patterns that is given by

$$\alpha_c = \frac{p}{N} = \sigma^2 \tag{2.2.24}$$

or approximately 0.185 for $\sigma = 0.430$. A more formal analysis, which we do not reproduce here,shows that there is a critical value of p/N at which the error fraction in a pattern jumps from about 1% to essentially no useful retrieval. The critical value $\alpha_c = 0.144$ ($\sigma = 0.379$) can be obtained from techniques developed in the study of spin glasses.

Question 2.2.1 Generalize the signal-to-noise analysis to describe the behavior of an initial pattern which is a Hamming distance B away from one of the imprinted patterns. Assume that the initial pattern is not correlated with any of the other imprinted patterns. Use the analysis to obtain an estimate of the basin of attraction of the imprinted patterns.

Solution 2.2.1 We initialize the network with a pattern that is a Hamming distance B from the first pattern. For convenience, we choose the pattern so that the first $N - B$ neurons are given by the first pattern,and the last B neurons are inverted:

$$s_i(0) = \begin{cases} \xi_i^1 & i \quad \{1,..., N-B\} \\ -\xi_i^1 & i \quad \{N-B+1,..., N\} \end{cases} \tag{2.2.25}$$

Our objective is to see whether the neural update will recover the imprinted pattern. To simplify the analysis, we assume that the recovery of the pattern must occur in the first update of the network. This will occur if the first $N - B$ neurons are stable and the last B neurons are unstable.

We check the stability of the first $N - B$ neurons by studying the stability of the first neuron. It is stable if

$$s_1(0)h_1 = \xi_1^1 h_1 = \xi_1^1 \sum_{j=2}^{N-B} J_{1j}\xi_j^1 + \xi_1^1 \sum_{j=N-B+1}^{N} J_{1j}\left(\xi_j^1\right) \tag{2.2.26}$$

is positive. Similarly, we check the stability of the last B neurons by studying the stability of the last neuron. It is unstable if:

$$s_N(0)h_N = \left(-\xi_N^1\right)h_N = \left(-\xi_N^1\right)\sum_{j=2}^{N-B} J_{Nj}\xi_j^1 + \left(-\xi_N^1\right)\sum_{j=N-B+1}^{N} J_{Nj}\left(-\xi_j^1\right) \tag{2.2.27}$$

is negative. Multiplying Eq. (2.2.27) by –1, we see that the condition that Eq.(2.2.27) is negative is actually the same as the condition that Eq.(2.2.26) is positive. Thus we can study the stability of the first neuron in order to verify whether the imprinted pattern will be recovered after one update.

Inserting the Hebbian form for the synapses after p imprints into Eq. (2.2.26), we have:

$$\xi_1^1 h_1 = \frac{1}{N}\sum_{j=2}^{N-B}\sum_{\mu=1}^{p}\xi_1^1\xi_1^\mu\xi_j^\mu\xi_j^1 - \frac{1}{N}\sum_{j=(N-B+1)}^{N}\sum_{\mu=1}^{p}\xi_1^1\xi_1^\mu\xi_j^\mu\xi_j^1 \tag{2.2.28}$$

We separate each sum in this expression into two parts. The first part is due to the imprint of the pattern we are trying to retrieve. The other part is due to all of the other patterns

$$\begin{aligned}\xi_1^1 h_1 &= \frac{1}{N}\sum_{j=2}^{N-B}\xi_1^1\xi_1^1\xi_j^1\xi_j^1 + \frac{1}{N}\sum_{j=2}^{N-B}\sum_{\mu=2}^{p}\xi_1^1\xi_1^\mu\xi_j^\mu\xi_j^1 \\ &\quad -\frac{1}{N}\sum_{j=(N-B+1)}^{N}\xi_1^1\xi_1^1\xi_j^1\xi_j^1 - \frac{1}{N}\sum_{j=(N-B+1)}^{N}\sum_{\mu=2}^{p}\xi_1^1\xi_1^\mu\xi_j^\mu\xi_j^1 \\ &= \frac{N-2B-1}{N} + \frac{1}{N}\sum_{j=2}^{N-B}\sum_{\mu=2}^{p}\xi_1^1\xi_1^\mu\xi_j^\mu\xi_j^1 - \frac{1}{N}\sum_{j=(N-B+1)}^{N}\sum_{\mu=2}^{p}\xi_1^1\xi_1^\mu\xi_j^\mu\xi_j^1\end{aligned} \tag{2.2.29}$$

where the parts due to the first imprinted pattern are explicitly evaluated because the square of either ±1 is 1. As before, the remaining sums constitute a random walk. It doesn't matter that there is a minus sign, since all of the terms are either ±1 and are uncorrelated. The total number of terms is $(N-1)(p-1)$. The typical magnitude (root mean square) of the noise is the same as before, but the signal term is different. The ratio of signal to noise is:

$$\frac{(N-2B)/N}{\sqrt{p/N}} = \frac{(N-2B)}{\sqrt{pN}} \tag{2.2.30}$$

where we have neglected 1 compared to both N and p.

To obtain an approximation to the size of the basin of attraction, we set the signal-to-noise ratio equal to the critical value for pattern stability $1/\sqrt{\alpha_c}$ obtained before. This gives a basin of attraction of size:

$$B = \frac{N}{2}\left(1-\sqrt{\frac{p}{\alpha_c N}}\right) \tag{2.2.31}$$

The result is consistent with two limiting results that we already know. The basin of attraction for a small number of imprints is just $N/2$, which is consistent with our discussion of a single imprint in Section 2.2.4. The basin of attraction vanishes when p reaches $\alpha_c N$. ∎

2.2.6 *Simulations*

The attractor neural network is well suited for simulation. In Fig. 2.2.5 we show the probability that an imprinted pattern is unstable and in Fig. 2.2.6 we show the number of stable imprinted patterns. Both are plotted as a function of the number of imprints p. The network used in the simulations has $N = 100$ neurons. The results are obtained by following the procedure:

1. Generate p random neural states $\{\xi_i^\mu\}$:

$$\xi_i^\mu = \pm 1 \qquad \mu = \{1,...,p\}, i = \{1,...,N\} \tag{2.2.32}$$

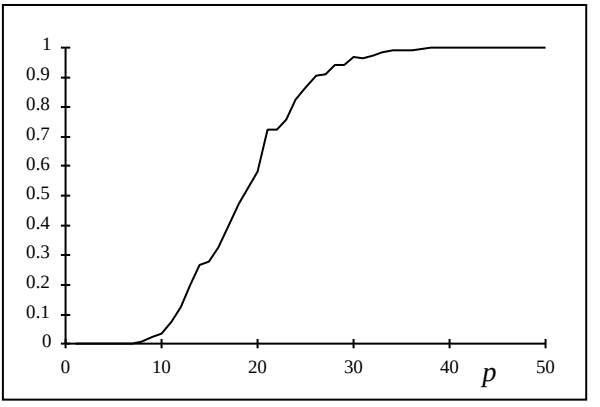

Figure 2.2.5 Fraction of unstable imprints as a function of the number of imprints p on a neural network of 100 neurons using Hebbian imprinting. For p less than 10 the stability of all of the stored patterns is perfect. Above this value the percentage of unstable patterns increases until all patterns are unstable. ∎

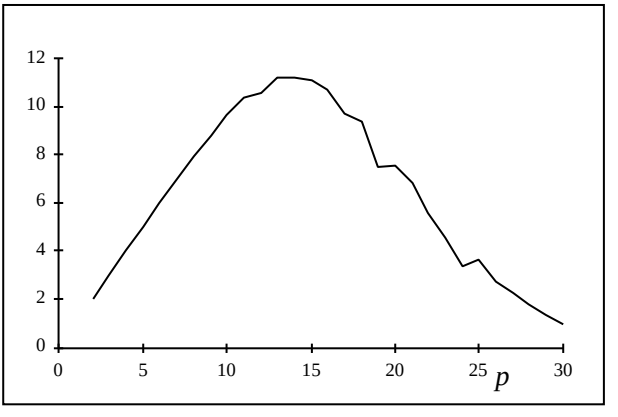

Figure 2.2.6 Number of stable imprints as a function of the number of imprints p on a neural network of 100 neurons using Hebbian imprinting. For p less than 10 all patterns are stable. The maximum number of stable imprinted patterns is less than 12. Above 15 imprints the number of stable patterns decreases gradually to zero. However, throughout this regime the basins of attraction of the patterns are very small and the system is not usable as a memory. ∎

2. Imprint them on the synapses of the neural network:

$$J_{ij} = \begin{cases} \dfrac{1}{N}\displaystyle\sum_{\mu=1}^{p} \xi_i^\mu \xi_j^\mu & i \neq j \\ 0 & i = j \end{cases} \tag{2.2.33}$$

3. Find the number of imprinted neural states that are stable:

$$p_{stable} = \sum_{\mu=1}^{p} \prod_{i} \delta\left(\xi_i^{\mu}, \operatorname{sign}\left(\sum_{j} J_{ij}\xi_j^{\mu}\right)\right) \tag{2.2.34}$$

where $\delta(i,j)$ is the Kronecker delta function defined by

$$\delta(i,j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{2.2.35}$$

4. Find the probability that a pattern is stable:

$$P_{stable} = p_{stable}/p \tag{2.2.36}$$

5. Average p_{stable} and P_{stable} over a number of trials (steps 1–4) with fixed N and p.

We can also investigate the basin of attraction. Consider a particular stable state of the neural network $\{s_i\}$ which may be an imprinted state. The basin of attraction of $\{s_i\}$ measures the size of the region of possible network states which is "attracted" to $\{s_i\}$. A neural state $\{s_i'\}$ is attracted to $\{s_i\}$ if $\{s_i'\}$ evolves to $\{s_i\}$ upon multiple application of the neuron update rule. Measuring the size of the basin of attraction is important, because the functioning of the neural network as an associative memory depends upon it.

In Fig. 2.2.7(a), the distribution of sizes of the basins of attraction B is shown for different numbers of imprints p on a network with 100 neurons. Each curve is normalized to 1 so that it gives the probability of finding a particular imprinted state with the specified basin of attraction. Fig. 2.2.7(b) shows the corresponding histograms of sizes of the basins of attraction for p imprinted states (the normalization of each curve is p). The maximum possible size of the basin of attraction is 50, half of the number of neurons. We can see from these figures that as the number of imprints increases, the average size of the basins of attraction decreases, and the width of their distribution increases. In addition, the number of imprinted states that are unstable increases. The algorithm used to obtain these figures includes the unstable states as having a basin of attraction of zero.

Fig. 2.2.7 was obtained using the following procedure:

1. Generate p random neural states $\{\xi_i^{\mu}\}$:

$$\xi_i^{\mu} = \pm 1 \qquad \mu = \{1,...,p\}, i = \{1,...,N\} \tag{2.2.37}$$

2. Imprint them on the synapses of the neural network:

$$J_{ij} = \begin{cases} \dfrac{1}{N}\displaystyle\sum_{\mu=1}^{p} \xi_i^{\mu}\xi_j^{\mu} & i \neq j \\ 0 & i = j \end{cases} \tag{2.2.38}$$

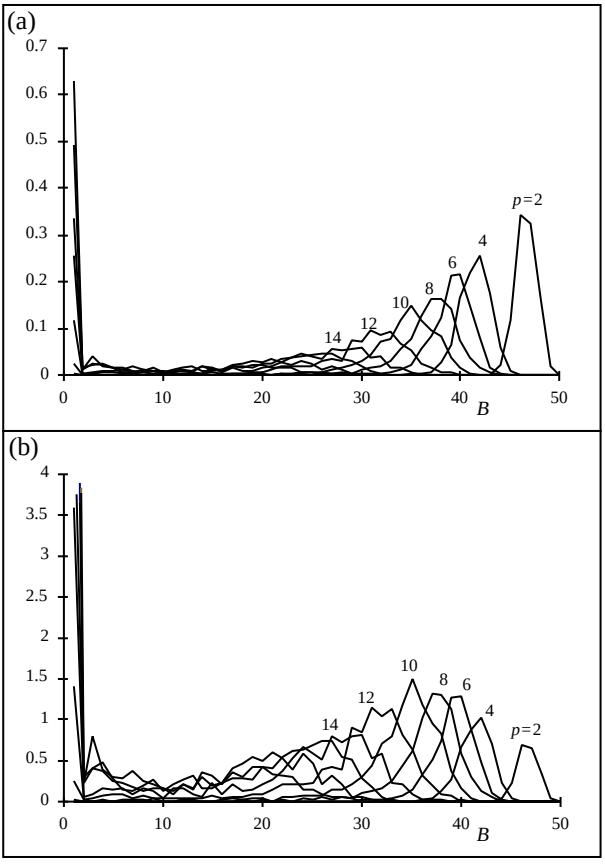

Figure 2.2.7 (a) Probability distribution of the size of the basin of attraction of imprinted patterns for a neural network of 100 neurons, and (b) histograms of the number of imprinted patterns with a particular basin of attraction. The horizontal axis is the Hamming distance, which measures the size of the basin of attraction. The probability distributions are normalized to 1, while the histograms are normalized to p. Each curve is for a different number of imprinted patterns as shown. The size of the basin of attraction decreases as the number of imprints increases. The probability distribution also broadens. When the number of imprints becomes greater than 10, the number of imprints with basins of attraction of zero begins to increase. This is the probability that a pattern is unstable as shown in Fig. 2.2.4. ∎

3. Find the basin of attraction of each of the imprinted patterns ξ_i^μ. The following steps measure the size of the basin of attraction by finding the average Hamming distance to a state which is not in the basin of attraction of the pattern ξ_i^μ:
   a. Set the neural state to ξ_i^μ.
   b. Pick (at random) an ordering of neurons $l(i)$.

c. Switch the state of each neuron in sequence according to the ordering $l(i)$ and find the minimum number of switches for which:the neural state resulting after at most a prespecified number n (taken to be 10) neural updates is *not equal* to the neural state ξ^μ:

$$w_j^{b,0} = \prod_{j'=1}^{b} (1 - 2\delta(j, l(j')))\xi_i^\mu \qquad b = \{0,...,N/2\}$$

$$w_i^{b,r} = \text{sign}\left(\sum_j J_{ij} w_j^{b,r-1}\right) \qquad r = \{1,...,n\} \tag{2.2.39}$$

$$B(\xi^\mu) = \min_{\prod_i \delta(\xi_i^\mu, w_i^{b,n}) = 0} b$$

The last expression specifies that we take the minimal value of b subject to the constraint that the state $w^{b,n}$ is not equal to ξ^μ. A straightforward procedure would increment b, evaluate $w^{b,n}$, and stop incrementing b when the condition of the last equation is satisfied.

d. Average over choices of neuron orderings $l(i)$.

4. Make a histogram of the basins of attraction for different ξ^μ, with fixed N and p.

Question 2.2.2 Use Glauber dynamics (Section 1.6.7) to introduce noise into the neural dynamics. Show that the noise actually can improve the retrieval of imprinted states. Specifically, use a network with 100 neurons and imprint 8 (random) neural states. Starting from a random neural state that was not imprinted, evolve the network a number of times with a measured amount of noise.Find the fraction of times that the network recovers one of the imprinted states. Vary the amount of noise to see its effect.

Why would noise increase the probability of retrieving the imprinted states? The imprinting not only creates basins of attraction for the imprinted states,it also causes the existence of many small shallow local energy minima that are called spurious memories. The noise enables the network to escape these shallow local energy minima and fall into the deeper energy minima that are the imprinted states. Spurious memories are discussed in Section 2.2.7.

Solution 2.2.2 Glauber dynamics is a standard implementation of a noisy update rule for neural networks. It uses a statistical rule for the state of each neuron at the next time step. At each time step the value of the neuron is set according to a probability given by the local field. The probability is written as:

$$P_{s_i}(+1;t) = \left(1 + \tanh\left(\beta \sum_i J_{ij} s_j(t-1)\right)\right)/2$$
$$P_{s_i}(-1;t) = 1 - P_{s_i}(+1;t) \tag{2.2.40}$$

where $\beta = 1/kT$ and T is the effective temperature associated with the noise. While Glauber dynamics uses asynchronous updating by selecting neurons to update at random,synchronous updating does not give significantly different results in many cases.

We can write the Glauber dynamics in an implicit way by introducing a temperature-dependent sign function which implies the probabilistic rule:

$$s_i(t) = \text{sign}_T\left(\sum_j J_{ij} s_j(t-1)\right) \tag{2.2.41}$$

where $\text{sign}_T(x)$ is suggestive of a finite temperature version of the sign function. However, this notation means nothing else than the previous probabilistic expression. In the limit as T approaches zero, the original $T = 0$ update rule is recovered. It should be noted that the temperature as used here does not necessarily correspond to the physical temperature. In the brain there is noisiness in the neural firing that may depend on the physical temperature, but may also be controlled by other factors.

With the introduction of the temperature-dependent update rule, care must be taken in defining the normalization of the synapse matrix J_{ij}. This is one of the reasons for the introduction of the normalization $1/N$ in the definition of the Hebbian imprinting rule (Eq. (2.2.6)).

Fig. 2.2.8 shows the probability of retrieving an imprinted state as a function of the amount of noise. This is the fraction of evolved-random states that result in imprinted states of the network (memories) for different values of β. No noise ($T = 0$) corresponds to $\beta = \infty$. The optimal noise for retrieval in these simulations is around $\beta = 4$. One problem in the simulations is how to identify when we have arrived at an imprinted state. Since we are evolving the network with noise, we should not arrive precisely at the imprinted state. One way to solve this problem is to assume that all states within a small Hamming distance are accepted as the imprinted state. For these simulations we avoid this problem by evolving the network at zero temperature,after it is evolved with noise.

Fig. 2.2.8 was generated using the following procedure:

1. Generate $p = 8$ random neural states $\{\xi_i^\mu\}$:

$$\xi_i^\mu = \pm 1 \qquad \mu = \{1,...,p\}, i = \{1,...,N\} \tag{2.2.42}$$

2. Imprint them on the synapses of the neural network:

$$J_{ij} = \begin{cases} \dfrac{1}{N}\displaystyle\sum_{\mu=1}^{p} \xi_i^\mu \xi_j^\mu & i \neq j \\ 0 & i = j \end{cases} \tag{2.2.43}$$

3. Generate a random neural state $\{w_i\}$:

$$w_i = \pm 1 \qquad i = \{1,...,N\} \tag{2.2.44}$$

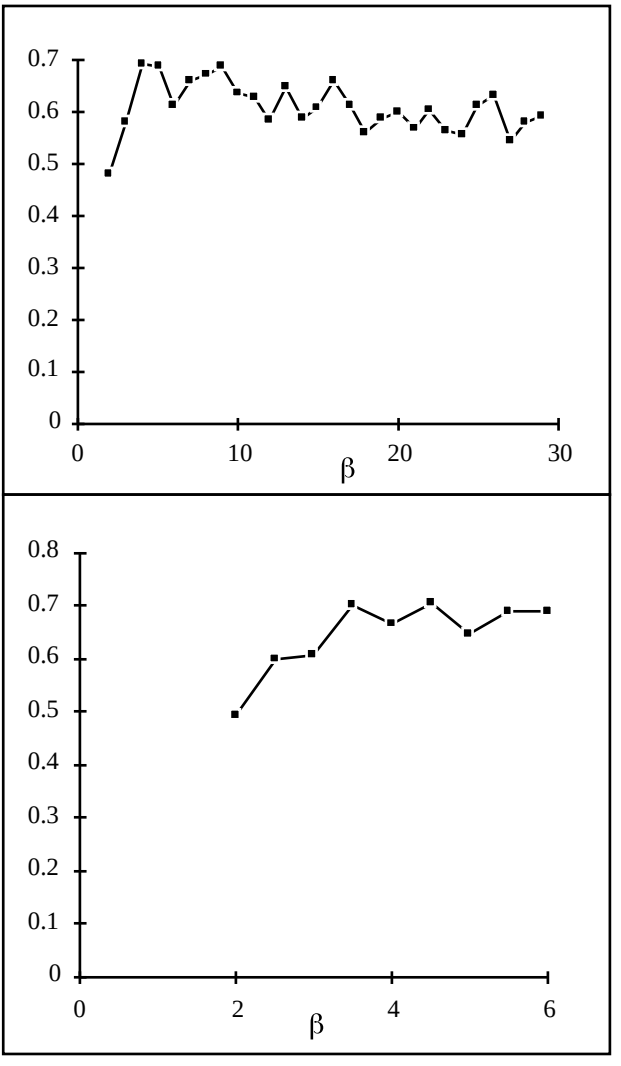

Figure 2.2.8 Tests of the influence of noise on the recovery of patterns imprinted on a network. The curves show the fraction of times neural evolution from a random initial state results in an imprinted state. The simulations use a network of $N = 100$ neurons and $p = 8$ imprinted states. The horizontal axis is the inverse of the effective temperature that describes the noise. The smallest amount of noise corresponds to the highest value of β. For low levels of noise the probability of recovering an imprinted state is less than 0.6. When noise is included the recovery rate can reach almost 0.7. The recovery rate improves gradually with increasing noise until about $\beta \approx 0.04$ when the recovery rate decreases dramatically. (a) shows a broader range of β, and (b) shows a narrower range near the optimal value of β for these simulations. The variability in the result, despite averaging in the simulations, reflects the importance of the particular (random) choice of imprinted patterns, and the use of only a limited number of updates of the network. ∎

4. Update the neural state $\{w_i\}$ according to the Glauber dynamics update rule $r_1 = 20$ times at temperature T:

$$w_i^r = \text{sign}_T\left(\sum_j J_{ij} w_j^{r-1}\right) \qquad r = \{1,...,r_1\}, i = \{1,...,N\} \tag{2.2.45}$$

5. Update the neural state $\{w_i^{r_1}\}$ according to the update rule $r_2 = 5$ times at $T = 0$:

$$w_i^r = \text{sign}\left(\sum_j J_{ij} w_j^{r-1}\right) \qquad r = \{r_1 + 1,...,r_1 + r_2\}, i = \{1,...,N\} \tag{2.2.46}$$

6. Find if the evolved neural state is equal to one of the originally imprinted states:

$$P_{stable} = \sum_\mu \prod_i \delta\left(\xi_i^\mu, w_i^{r_1+r_2}\right) \tag{2.2.47}$$

7. Average P_{stable} over different trials to find the proportion of evolved random states equal to one of the imprinted states. ∎

2.2.7 *Overload and spurious states*

We have discussed the storage capacity of attractor networks with Hebbian imprinting. As part of this discussion we showed that the basins of attraction of the imprinted states go to zero when the memory becomes overloaded. This implies that there is a catastrophic failure of the network—when we exceed capacity, all of the memories are forgotten. The reason that this occurs is that all of the memories are treated the same by the imprinting process. When the capacity is exceeded, there is no mechanism for the network to select which of them to remember.

There are modifications of the imprinting rule that enable the memory to retain some of the imprinted patterns as memories, at the expense of losing the others. The simplest way to determine which imprints to remember is by the order of the imprint. Rather than keeping the first few imprints, it makes sense to retain the most recent (last few) imprints. A memory that retains the most recent imprints is known as a palimpsest memory, after the name of parchments that were erased and reused in medieval times. Historians benefited from the residuals of earlier writings that remained visible. For our neural network implementation, we could modify the Hebbian imprinting by progressively increasing the strength of the imprint of patterns by a factor $e^{\varepsilon/N}$:

$$J_{ij}(t) = J_{ij}(t-1) + e^{+\varepsilon t/N} \xi_i^t \xi_j^t \qquad (i \neq j) \tag{2.2.48}$$

where we assume that each imprint is performed in a unit time interval, and the patterns are indexed by time. A value of $\varepsilon = 8.44$ has been calculated as optimal for storing random neural states. In general this and other palimpsest memories reduce the effective capacity of the network. Because of the difference in treatment of recent ver-

sus older memories, there is a significant degradation in total number of memories retained. This sacrifice occurs for the benefit of ensuring that some memories are retained after overload would otherwise occur.

Note that multiplying all the synapses J_{ij} by a constant does not affect any results of retrieval or stability. Thus, only the relative strength of different imprints is important. If a bound on the magnitude of synapses is desired, we can adopt the expression

$$J_{ij}(t) = e^{-\varepsilon/N} J_{ij}(t-1) + \xi_i^t \xi_j^t \qquad (i \neq j) \tag{2.2.49}$$

instead of Eq. (2.2.48). This is more like the erasure of previous writing, because all the synapses are reduced by the factor $e^{-\varepsilon/N}$ before the next imprint.

While there are methods,like these palimpsest memories, to ensure that overload does not occur, it is important to understand how overload occurs. Overload is a natural mode of failure of the attractor neural network. Therefore,it is likely to occur for biological networks under some circumstances. Detailed studies of attractor networks at high capacity indicate that the behavior of the network near overload becomes dominated by what are called spurious memories. We have spoken about the imprinted patterns as if they are the only stable states of a network. This is not the case. Spurious memories are stable states of the network that were not imprinted. Without an independent way of telling whether they were imprinted or not, these states masquerade as memories, but they are not. Spurious memories are not completely unrelated to the imprinted states. Instead they are generally a mixture of states. One example is a state formed by a majority rule from three imprinted states:

$$s_i = \text{sign}(\xi_i^1 + \xi_i^2 + \xi_i^3) \tag{2.2.50}$$

In Question 2.2.3 the stability of this state is shown using a signal-to-noise analysis. The problem that arises as the number of imprints increase is that the number of such spurious states increases combinatorially (greater than exponentially) with the number of imprints. The growth in the number of spurious states occurs because they are formed from all possible combinations of the imprinted states. When overload occurs,it is actually the basins of attraction of these states that swamp the basins of attraction of the imprinted states.

Once the spurious states swamp the imprinted states, the network becomes essentially equivalent to a spin glass (Section 1.6) that has random weights for each of the synapses. We can understand this qualitatively because the noise becomes larger than the signal in the signal-to-noise analysis. Thus the energy of any state is given by a sum over random variables. Beyond overload, the characteristics of the neural network become similar to those of the spin glass, where there are a hierarchically structured set of minimum energy configurations with large barriers between them. The lowest energy states are not the imprinted ones.

Question 2.2.3 Evaluate the stability of the symmetric mixture of three states given by Eq. (2.2.50) using a signal-to-noise analysis. Hint: convince yourself that the noise is essentially the same as that for an imprinted state and evaluate only the signal.

Solution 2.2.3 A difficulty in studying the stability of the spurious pattern given by Eq. (2.2.50) is that there are two distinct types of neurons—those where all three patterns in the mixture have the same activity, and those where only two out of the three have the same activity. It is important to distinguish these two cases. We evaluate first the signal and then discuss the noise.

The stability of s_1 for the state given by Eq. (2.2.50) is determined by

$$s_1 h_1 = \text{sign}(\xi_1^1 + \xi_1^2 + \xi_1^3) \sum_{j=2}^{N} J_{1j} \text{sign}(\xi_j^1 + \xi_j^2 + \xi_j^3) \tag{2.2.51}$$

Inserting the Hebbian form for the synapses after p imprints gives

$$s_1 h_1 = \frac{1}{N} \sum_{j=2}^{N} \sum_{\mu=1}^{p} \text{sign}(\xi_1^1 + \xi_1^2 + \xi_1^3) \xi_1^\mu \xi_j^\mu \text{sign}(\xi_j^1 + \xi_j^2 + \xi_j^3) \tag{2.2.52}$$

Our objective is to determine the probability that this is negative, in which case s_1 is unstable. As in the treatment of the imprinted patterns in the text, we do this by evaluating the average (the signal) and the standard deviation (the noise) of the distribution, and approximate the distribution as a Gaussian.

The signal arises from the first three terms in the sum over μ, which we separate to obtain:

$$\begin{aligned} s_1 h_1 = & \frac{1}{N} \sum_{j=2}^{N} \text{sign}(\xi_1^1 + \xi_1^2 + \xi_1^3) \left(\xi_1^1 \xi_j^1 + \xi_1^2 \xi_j^2 + \xi_1^3 \xi_j^3 \right) \text{sign}(\xi_j^1 + \xi_j^2 + \xi_j^3) \\ & + \frac{1}{N} \sum_{j=2}^{N} \sum_{\mu=4}^{p} \text{sign}(\xi_1^1 + \xi_1^2 + \xi_1^3) \xi_1^\mu \xi_j^\mu \text{sign}(\xi_j^1 + \xi_j^2 + \xi_j^3) \end{aligned} \tag{2.2.53}$$

The average value of the first sum depends on whether ξ_1^1, ξ_1^2 and ξ_1^3 have the same sign. If they do we have a signal given by:

$$\langle s_1 h_1 \rangle = \frac{1}{N} \sum_{j=2}^{N} \left(\xi_j^1 + \xi_j^2 + \xi_j^3 \right) \text{sign}(\xi_j^1 + \xi_j^2 + \xi_j^3) = 3 \times \frac{1}{4} + 1 \times \frac{3}{4} = \frac{3}{2} \tag{2.2.54}$$

where the intermediate equation indicates the value of the term multiplied by the probability of its occurrence. Thus, terms that have a magnitude of 3 occur 1/4 of the time, while terms that have a magnitude of 1 occur 3/4 of the time. Similarly, if ξ_1^1, ξ_1^2 and ξ_1^3 do not have the same sign, then two out of three of them have the same sign, and they have a signal given by:

$$\langle s_1 h_1 \rangle = \frac{1}{N} \sum_{j=2}^{N} \left(\xi_j^1 + \xi_j^2 - \xi_j^3 \right) \text{sign}(\xi_j^1 + \xi_j^2 - \xi_j^3) = 1 \times \frac{1}{4} + 3 \times \frac{1}{4} - 1 \times \frac{1}{2} = \frac{1}{2} \tag{2.2.55}$$

We see that for the first type of neuron (1/4 of the neurons are of this type) the signal is higher than the signal of an imprinted pattern. On the other hand for the second type of neuron (the remaining 3/4 of the neurons) the signal is lower than that of an imprinted pattern.

The noise can be determined by direct evaluation of the standard deviation of Eq.(2.2.53). However, we can convince ourselves that it is not much different than the noise found for an imprinted pattern in Eq.(2.2.22). The last sum in Eq. (2.2.53) is a sum over $(N-1)(p-3)$ uncorrelated random values of ± 1. Its root mean square magnitude is $\sigma_2 = \sqrt{(p-3)/N}$. This is most of the noise for $p >> 1$ because the first sum,after we subtract the mean, contains no more than $3N$ uncorrelated terms with magnitude one. Thus the standard deviation of the first sum is no more than roughly $\sigma_1 \approx \sqrt{3/N}$, and the total standard deviation satisfies

$$\sigma^2 = \sigma_1^2 + \sigma_2^2 \approx \sigma_2^2 \approx \sqrt{p/N} \tag{2.2.56}$$

for $p >> 1$. This is the same as the noise term found for imprinted patterns.

The main conclusion that we reach from this analysis is that for low storage, $p << N$, the neurons have a signal that is much greater than the noise, so the pattern will be stable. The observation above that 3/4 of the neurons have a signal that is half of the signal in an imprinted pattern implies that the basin of attraction of the spurious patterns is shallower and smaller than that of the imprinted patterns. ∎

2.3 Feedforward Networks*

2.3.1 *Defining feedforward networks*

Feedforward networks (Fig. 2.3.1) are convenient for visualizing input-output systems. They have also been more extensively used in the construction of commercial applications than other neural network models.A feedforward network is composed of several layers. The number of these layers is not large,in part because of difficulties in training these networks. The synapses of a feedforward network are unidirectional. The neuron activity is represented by a continuous variable over a limited range of possible values. We take the range of values to be (−1,+1):

$$s_i^l \quad (-1,+1) \qquad\qquad i \quad \{1,...,N_l\}, l \quad \{1,...,L\} \tag{2.3.1}$$

where l is the layer index,and the number of neurons in a layer N_l may vary from layer to layer. For the synapses we adopt the notation:

$$J_{ij}^l \qquad\qquad i \quad \{1,...,N_{l+1}\}, j \quad \{1,...,N_l\}, l \quad \{1,...,L-1\} \tag{2.3.2}$$

For L layers of neurons there are $L-1$ sets of synapses. By our indexing conventions, the last set of synapses is J_{ij}^{L-1}.

*This section may be omitted without significant loss of continuity.

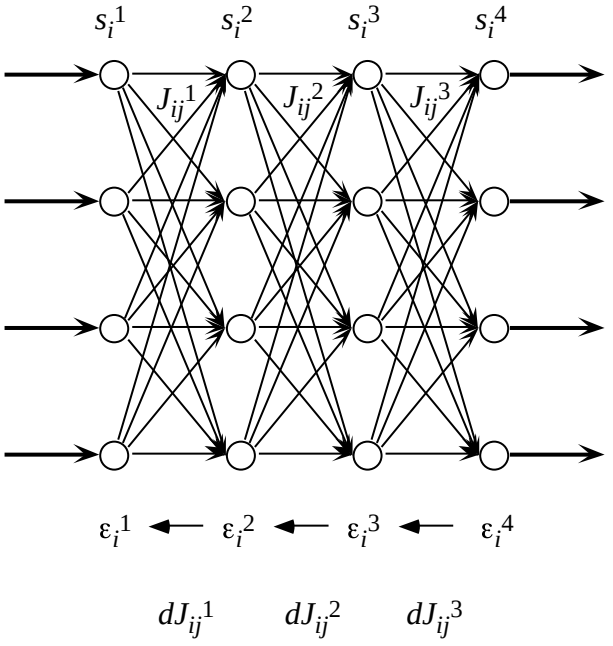

Figure 2.3.1 Schematic of a feedforward network showing the notation for the neurons s_i^l and synapses J_{ij}^l running in only one direction between the layers of neurons. The input to the network enters from the left and the output is read from the right. The most commonly used training algorithm for the feedforward network is the back-propagation algorithm. Starting from an initial set of values of the synapses, the output is calculated from a preselected input for which a desired output is known. The desired output is compared with the output that is calculated. The difference is the error on the last layer of neurons (here the fourth layer) ε_i^4. The error is used to change incrementally the synapses J_{ij}^3 so as to reduce the error. The error is also used to obtain the corresponding error on the previous layer ε_i^3 so that the previous layers of synapses J_{ij}^2 can also be corrected accordingly. In this way the error is propagated backward through the network to correct all of the synapses. ∎

The propagation of input through the network proceeds in a layer by layer fashion. We can picture this as a signal that is propagating through the network, so that the layer index l in the neuron variable s_i^l becomes the analog of a time index. However, it is important to recognize that the propagation through the network is both the space and time coordinate when the processing of an individual input pattern is considered. There is no other time coordinate in the network operation.

The update rule that determines the activity of a neuron at a particular time is a function of the influence of the neurons of the previous layer, usually taken to be sigmoidal:

$$s_i^{l+1} = \mathrm{sgm}\left(\sum_j J_{ij}^l s_j^l\right) \tag{2.3.3}$$

The sigmoidal function may be the function $\tanh(x)$, which may also be generalized to

$$\mathrm{sgm}(x) = \tanh(\beta(x - h)) \tag{2.3.4}$$

The parameter β is an overall multiplier of the synaptic weights, and therefore is redundant. It may nevertheless be convenient to use it under some circumstances. h is an additional parameter that could vary from neuron to neuron (with the notation h_i^l), and may also be adjusted as part of the training procedure described in the following section. Other forms of sigmoidal function may be used as well.

2.3.2 *Operating and training feedforward networks*

The operation of a feedforward network begins with the imposition of a pattern of activity on the first layer of the neurons. This pattern is assumed to represent the operation of sensory neurons. The activity of each successive layer is then determined according to Eq. (2.3.3). The action of the network on this input ends with the extraction of the neuron activities from the final layer of neurons. This extraction may be considered to be an effect—an action caused by motor neurons. Alternatively, we could consider the activities of the final layer of neurons to be a new representation of the input sensory information—the recognition of a pattern.

To train the network synapses, we begin from a set of examples of input and output pairs that the network should emulate. The objective is to produce the specified outputs from the specified inputs. Once the network is trained, as far as possible, to produce the desired output from each input, it automatically generalizes from these training examples. The generalization is obtained by inputting other patterns to the network and obtaining the resulting output. In effect, the network interpolates between the training examples.

We designate the input and output training pairs (of which there are p) as:

$$(\xi_i^\nu, \eta_j^\nu) \qquad i \in \{1,\ldots,N_1\}, j \in \{1,\ldots,N_L\}, \nu = \{1,\ldots,p\} \tag{2.3.5}$$

The training of the feedforward network can be performed in many ways. The most common method begins from the recognition that it is only the values of the neurons in the final layer that explicitly matter to the operation of the system. The layers between the input and output layer are "hidden." The objective is to optimize the agreement between the action of the network and the desired output. To achieve this we write a cost function (energy), which measures the error—the difference between the value of the output neurons after action on the input trial state and the desired output. The cost function is:

$$E[\{J_{ij}^l\}] = \sum_{\nu=1}^{p} \sum_{i=1}^{N_L} (s_i^L(\nu) - \eta_i^\nu)^2 \tag{2.3.6}$$

where we have introduced the notation $s_i^L(\nu)$ to indicate the activities of the Lth layer of neurons that result from application of the network to the νth input. For simplicity, the different errors are weighted equally. Implicitly, $s_i^L(\nu)$ and the cost function depend on the values of all of the synaptic variables J_{ij}^l. The cost function should be min-

imized with respect to them. The cost function may be minimized in a variety of ways (Section 1.7.4), however, as usual for a problem with a high-dimensional minimization space, there may be problems with many local minima. Nevertheless, the simplest approach is a steepest-descent minimization algorithm.

Before proceeding with a mathematical derivation of the most common approach to minimizing the cost function, we briefly summarize the results. At each step of the procedure, we present to the network a particular one of the input examples. Once the network has operated on the input, we compare the activities of the last neuron layer to the desired output. Our objective is to decrease the error. The easiest way to improve the agreement is to change the last layer of synapses. We calculate the direction to change these synapses to improve the agreement. In general, making this change in the synapses will not be sufficient. To improve the agreement further, we take the error at each of the output neurons and determine what changes would be needed in the activity of the previous layer of neurons in order to correct the output-neuron values. This is done using the existing synapses between the two layers. This step, taking the error in the final layer of neurons and identifying the corresponding error in the previous layer of neurons, is called back-propagation of error. Once we know the error in the second to last layer of neurons, we can find the direction to change the second to last layer of synapses. We repeat the procedure for earlier layers, and correct incrementally all of the synapses of the network.

The following derivation is difficult only because of the number of subscripts and superscripts. We adopt standard practice and assume that we will minimize the cost function by modifying J_{ij}^{l} in steps that reduce the cost function successively for each of the patterns separately. Convergence is not guaranteed, but will work if the cost function is well behaved. We thus adopt the partial cost function

$$E^{\nu}[\{J_{ij}^{l}\}] = \sum_{i=1}^{N_L} (s_i^L(\nu) - \eta_i^{\nu})^2 \tag{2.3.7}$$

To minimize this function we change J_{ij}^{l} in the direction of steepest descent:

$$\begin{aligned} J_{ij}^{l}(t+1) &= J_{ij}^{l}(t) + \delta J_{ij}^{l}(t) \\ \delta J_{ij}^{l}(t) &= -c\,\frac{\partial E^{\nu}[\{J_{ij}^{l}\}]}{\partial J_{ij}^{l}(t)} \end{aligned} \tag{2.3.8}$$

We use the time variable to indicate repetitive cycling over the different patterns ν. It keeps track of the steps in the minimization, not propagation of the signal through the network. c is chosen small enough, and possibly time-dependent, to provide for convergence. In principle it doesn't matter in which order we consider the J_{ij}^{l}, but it is convenient to start from the synapses leading to the final (Lth) layer J_{ij}^{L-1}.

$$\delta J_{ij}^{L-1}(t) = -c\,\frac{\partial E^{\nu}[\{J_{ij}^{l}\}]}{\partial J_{ij}^{L-1}(t)} = -c\,\frac{\partial \sum_{k=1}^{N_L} (s_k^L(\nu) - \eta_k^{\nu})^2}{\partial J_{ij}^{L-1}(t)} \tag{2.3.9}$$

where we have taken care to use a new index for the sum in the numerator. Taking the derivative inside the sum we have:

$$\delta J_{ij}^{L-1}(t) = -2c \sum_{k=1}^{N_L} (s_k^L(\nu) - \eta_k^\nu) \frac{\partial s_k^L(\nu)}{\partial J_{ij}^{L-1}(t)} \tag{2.3.10}$$

The value of the kth neuron in the Lth layer can only depend on synapses leading directly to it, and not on other synapses. Thus the derivative has to be zero unless $k = i$, and the other terms in the k sum can be neglected. We show this explicitly using the expression for $s_k^L(\nu)$ in terms of the previous layer of neurons:

$$\begin{aligned}
\frac{\partial s_k^L(\nu)}{\partial J_{ij}^{L-1}(t)} &= \frac{\partial\, \mathrm{sgm}(\sum_m J_{km}^{L-1}(t) s_m^{L-1}(\nu))}{\partial J_{ij}^{L-1}(t)} \\
&= \mathrm{sgm}'(\sum_m J_{km}^{L-1}(t) s_m^{L-1}(\nu)) \sum_n s_n^{L-1}(\nu) \frac{\partial J_{kn}^{L-1}(t)}{\partial J_{ij}^{L-1}(t)} \\
&= \mathrm{sgm}'(\sum_m J_{km}^{L-1}(t) s_m^{L-1}(\nu)) \sum_n s_n^{L-1}(\nu) \delta_{k,i} \delta_{n,j} \\
&= \mathrm{sgm}'(\sum_m J_{km}^{L-1}(t) s_m^{L-1}(\nu)) s_j^{L-1}(\nu) \delta_{k,i}
\end{aligned} \tag{2.3.11}$$

The prime on the sigmoidal function has the conventional meaning of a derivative with respect to its argument. In the third line we made use of the knowledge that $s_m^{L-1}(\nu)$ is independent of J_{ij}^{L-1} because the $(L-1)$st layer of neurons precede the $(L-1)$st layer of synapses.

Returning to the evaluation of the change in J_{ij}^{L-1} we find

$$\begin{aligned}
\delta J_{ij}^{L-1}(t) &= -2c \sum_{k=1}^{N_L} (s_k^L(\nu) - \eta_k^\nu) \mathrm{sgm}'(\sum_m J_{km}^{L-1}(t) s_m^{L-1}(\nu)) s_j^{L-1}(\nu) \delta_{k,i} \\
&= -2c (s_i^L(\nu) - \eta_i^\nu) \mathrm{sgm}'(\sum_m J_{im}^{L-1}(t) s_m^{L-1}(\nu)) s_j^{L-1}(\nu)
\end{aligned} \tag{2.3.12}$$

We can simplify the notation by defining two auxiliary quantities. We define the error at the Lth layer as:

$$\varepsilon_i^L(\nu) = (s_i^L(\nu) - \eta_i^\nu) \tag{2.3.13}$$

The derivative of the sigmoidal function in Eq. (2.3.12) could be written as a function of the neuron $s_i^L(\nu)$, since it applies the sigmoidal derivative to the same argument (the postsynaptic potential) that determines the neuron value. We call this function $w(s)$:

$$w(s_i^L(\nu)) = \mathrm{sgm}'(\sum_m J_{im}^{L-1}(t) s_m^{L-1}(\nu)) = \mathrm{sgm}'(\mathrm{sgm}^{-1}(s_i^L(\nu))) \tag{2.3.14}$$

This leads to the simplified form of Eq. (2.3.12)

$$\delta J_{ij}^{L-1}(t) = -2c\, \varepsilon_i^L(\nu) w(s_i^L(\nu)) s_j^{L-1}(\nu) \tag{2.3.15}$$

which is the desired result.

Before we continue to find the incremental changes in the other J_{ij}^{l} for $l < L-1$, we discuss the result for J_{ij}^{L-1} in a simple case. Consider Eq. (2.3.15) if all of the neurons have a level of activity that is within the linear range of the sigmoidal function, taken to be $\tanh(x)$. This would be equivalent to neglecting the nonlinear response of the neurons. Then $\mathrm{sgm}'(x) = 1$ and $w(s_i^L(\nu)) = 1$. Inserting Eq.(2.3.15) into Eq.(2.3.8) we have:

$$J_{ij}^{L-1}(t+1) \cong J_{ij}^{L-1}(t) - 2c(s_i^L(\nu,t) - \eta_i^\nu)s_j^{L-1}(\nu,t) \quad \text{(linear regime)} \tag{2.3.16}$$

Note that all of the neuron activities, indexed by ν, also have a t index,since they depend on the synapses, and thus their values change during the minimization. We can act with the new synapse values on the neurons of the previous layer to obtain the new neuron values in the final layer. We assume that the values of the neurons at the previous layer have not been changed. This would be true if we chose not to modify the previous layers of synapses, or if there were only two layers of neurons (one layer of synapses), then:

$$s_i^L(\nu, t+1) \cong \sum_j J_{ij}^{L-1}(t+1)s_j^{L-1}(\nu,t) \tag{2.3.17}$$

Inserting Eq. (2.3.16)

$$\begin{aligned} s_i^L(\nu,t+1) &\cong \sum_j J_{ij}^{L-1}(t)s_j^{L-1}(\nu,t) - 2c(s_i^L(\nu,t) - \eta_i^\nu)\sum_j s_j^{L-1}(\nu,t)s_j^{L-1}(\nu,t) \\ &\cong s_i^L(\nu,t) - 2c(s_i^L(\nu,t) - \eta_i^\nu)\sum_j s_j^{L-1}(\nu,t)s_j^{L-1}(\nu,t) \end{aligned} \tag{2.3.18}$$

From this we see that if the neuron values of layer $L-1$ are normalized to 1 ($\sum s_j^{L-1}s_j^{L-1} = 1$) and $c = 1/2$ (or if c is chosen to be 1/2 of the inverse of the normalization) then convergence will be perfect for the pattern ν, since then

$$s_i^L(\nu, t+1) \cong \eta_i^\nu \tag{2.3.19}$$

More generally, a smaller value of c will bring the neuron values closer to the desired result, as should be expected from a steepest descent. This shows that for a single input-output training pair, the cost function may be readily minimized using only the linear regime of one layer of synapses. Constructing a network that will perform a desired pattern-recognition task can be much more difficult when there are many input-output training patterns representing the task.

We return to the main line of our discussion and consider the second to last layer of synapses J_{ij}^{L-2}

$$\delta J_{ij}^{L-2}(t) = -c\frac{\partial E^\nu[\{J_{ij}^l\}]}{\partial J_{ij}^{L-2}(t)} = -c\sum_k \frac{\partial E^\nu[\{J_{ij}^l\}]}{\partial s_k^L(\nu)}\frac{\partial s_k^L(\nu)}{\partial s_i^{L-1}(\nu)}\frac{\partial s_i^{L-1}(\nu)}{\partial J_{ij}^{L-2}(t)} \tag{2.3.20}$$

The latter expression uses the sequentiality of the determination of the neuron values. We have also taken into account that in the $(L-1)$st layer, it is only the ith neuron that depends on the synapse J_{ij}^{L-2}. Each of the factors is readily evaluated:

$$\frac{\partial E^{\nu}[\{J_{ij}^{l}\}]}{\partial s_k^L(\nu)} = 2(s_k^L(\nu) - \eta_k^{\nu}) = 2\varepsilon_k^L(\nu)$$

$$\frac{\partial s_k^L(\nu)}{\partial s_i^{L-1}(\nu)} = \frac{\partial\, \mathrm{sgm}(\sum_m J_{km}^{L-1}(t) s_m^{L-1}(\nu))}{\partial s_i^{L-1}(\nu)} = \mathrm{sgm}'(\sum_m J_{km}^{L-1}(t) s_m^{L-1}(\nu)) J_{ki}^{L-1}(t)$$
$$= w(s_k^L(\nu)) J_{ki}^{L-1}(t) \tag{2.3.21}$$

$$\frac{\partial s_i^{L-1}(\nu)}{\partial J_{ij}^{L-2}(t)} = \frac{\partial\, \mathrm{sgm}(\sum_n J_{in}^{L-2}(t) s_n^{L-2}(\nu))}{\partial J_{ij}^{L-2}(t)} = \mathrm{sgm}'(\sum_n J_{in}^{L-2}(t) s_n^{L-2}(\nu)) s_j^{L-2}(\nu)$$
$$= w(s_i^{L-1}(\nu)) s_j^{L-2}(\nu)$$

leading to:

$$\delta J_{ij}^{L-2}(t) = -2c \sum_k \varepsilon_k^L(\nu) w(s_k^L(\nu)) J_{ki}^{L-1}(t) w(s_i^{L-1}(\nu)) s_j^{L-2}(\nu) \tag{2.3.22}$$

We can recast this expression into the same form as for the synapses J_{ij}^{L-1}

$$\delta J_{ij}^{L-2}(t) = -2c\, \varepsilon_i^{L-1}(\nu) w(s_i^{L-1}(\nu)) s_j^{L-2}(\nu) \tag{2.3.23}$$

by defining the error on the $(L-1)$st layer of neurons as:

$$\varepsilon_i^{L-1}(\nu) = \sum_k \varepsilon_k^L(\nu) w(s_k^L(\nu)) J_{ki}^{L-1}(t) \tag{2.3.24}$$

This expression,in effect,takes the error that was known on the Lth layer and obtains the error on the previous layer using the existing set of synapse values $J_{ki}^{L-1}(t)$. This procedure is known as back-propagation of error and gives the name back-propagation algorithm to this method of training feedforward networks.

The modification of earlier layers is obtained similarly by extending this analysis layer by layer. In each case, an expression of the form of Eq. (2.3.24) can be written that takes the error of one layer and sends it back to the previous layer. The correction to the synapses is then written as in Eq. (2.3.23).

2.4 Subdivided Neural Networks

Among the objectives of the study of neural networks is the development of a basis for an understanding of sensory processing, motor control, memory and higher information-processing functions of the brain. In previous sections, we have seen that it is possible to describe an associative content-addressable memory using an attractor neural network. The associative memory captures an important generic property that we would like to build upon to understand additional aspects of brain function. We also touched upon some aspects of the processing by feedforward networks that are suggestive of sensory-motor systems. However, most of the higher information-

processing tasks, of which the brain is readily capable, appear remote from these considerations.

In order to make progress in understanding the higher information-processing functions of the brain we must construct an additional level of organization between neurons and the brain—that of brain subdivisions or subnetworks. The substructure of the brain is well known to students of neurophysiology (see Fig. 2.4.1). Experimentally, the mapping of brain function has identified sensory- or motor-related aspects of the brain—visual processing centers, auditory processing centers, the motor cortex, as well as aspects of language processing that may be counted among the higher information-processing functions. There is a long-standing debate regarding the degree of localization of function in the brain. In a simplified form, the debate is between two camps, one suggesting that specific functions are localized on individual neurons, the other suggesting complete delocalization of function throughout the brain. At present, experimental evidence has led to general agreement that at least an intermediate degree of regional specialization exists.

As we discussed in Section 1.10.7, most fields of inquiry are built upon levels of description each of which is constructed on finer scales. To neglect the description of the subdivisions of the brain and try to explain brain function directly from the behavior of individual neurons would be to skip an important and simplifying level of description. One of our primary tasks, therefore, in studying neural networks, is to investigate and identify the function and interaction of subnetworks. We hope then to build models of human information-processing using subnetworks as the analog of brain subdivisions. It is almost a separate endeavor to construct such a theory of higher information-processing. The historical efforts in this area are the theories of the mind such as those of Freud, which separated the mind in two ways. The first separation was between the conscious and subconscious and the second between the id, ego and superego. This and other theoretical models should be considered within the domain of our inquiry. However, until we have a better understanding of the function of brain subdivisions, we will not be able to evaluate the validity of the many psychological theories or propose more complete ones.

There are two forms of subdivision that can be readily identified—longitudinal and lateral. We have already considered the longitudinal form of subdivision in the example of feedforward networks. A multilayer feedforward network describes a set of neural subdivisions each of which is a single layer of the network. In this model there are no synapses within a neural subdivision, all of the synapses run between subdivisions, specifically in a feedforward direction. The input layer (or first few layers) represents sensory processing, and the output layer (or last few layers) represents motor control. Intermediate layers are less clearly identified. This longitudinal subdivision in feedforward networks is directly related to sequential stages in processing. Longitudinal subdivision in feedforward networks is necessary because of limitations on what a single layer of synapses can be trained to accomplish.

In the remainder of this chapter we consider the second type of subdivision—lateral subdivision—formed when the synaptic connections within each subdivision are of greater number or of greater strength than between the subdivisions. In contrast to

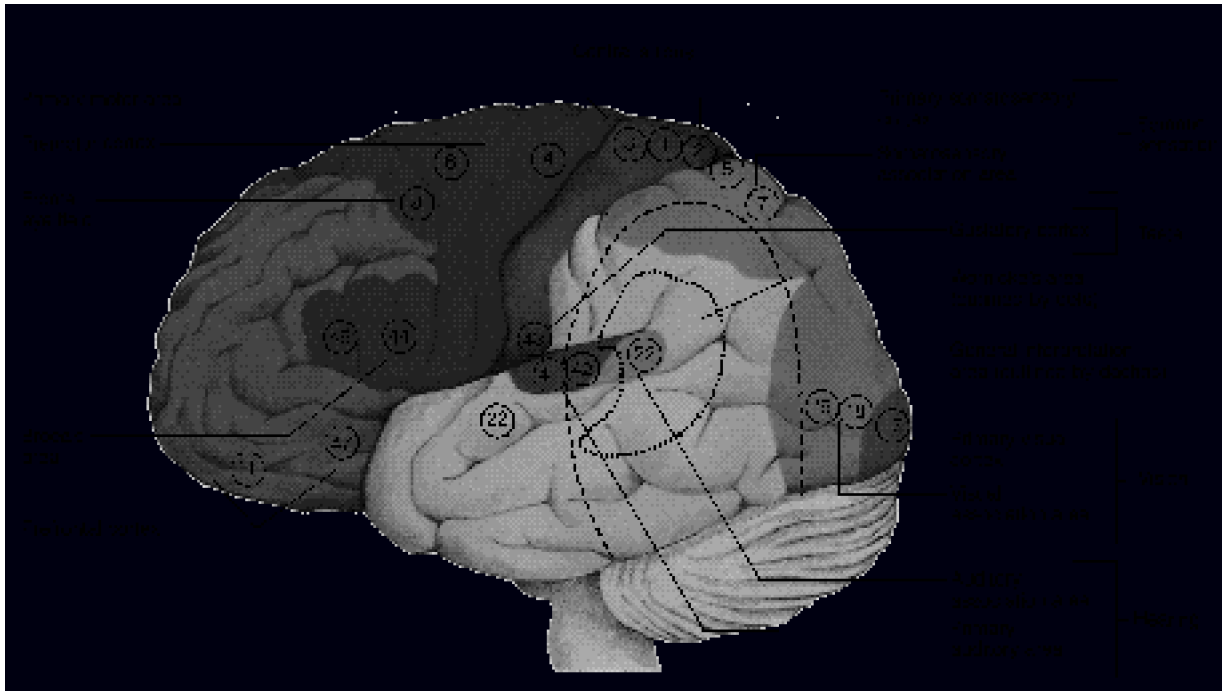

Figure 2.4.1 Illustration of the functional anatomy of the brain as determined by neurophysiological experiments (from E. N. Marieb, *Human Anatomy and Physiology*, [Benjamin Cummings, 1992]) ▮

longitudinal subdivision, lateral subdivision separates the processing of sensory or other information into parallel channels. This kind of subdivision can be treated and understood within the attractor network model (Fig. 2.4.2).

In developing an understanding of the role of subdivisions in the brain, we must begin from basic questions. The most basic is the question, Why should the brain be subdivided at all? This may seem a simple question,since it might seem obvious that, for example, language should be separate from visual processing and from auditory processing and from motor control—doesn't this make sense? But we know that all of these are also connected to each other. Why then should we not process them all together? In the attractor network, we simplify the consideration of network function to that of an associative memory. If we compare a subdivided attractor network with the fully connected attractor network, we immediately run into a fundamental problem—a lower storage capacity.

The storage capacity of a neural network increases with the degree of interconnectedness. In Section 2.2 we determined the storage capacity for the fully connected network with Hebbian imprinting. The network could store $\alpha_c N$ patterns. We can count,instead,the total number of independent bits in the stored patterns. Since each pattern has N bits,this gives a total of $\alpha_c N^2$ bits. This is somewhat deceptive,since in the limit of maximum storage, just below overload, we must present almost all of the pattern in order for it to be "retrieved." However, because any part of the pattern could be retrieved, we might still consider $\alpha_c N^2$ to be the maximum number of bits of information stored in the network. The expression $\alpha_c N^2$ should not be surprising, since the information is stored in the synapses and the number of synapses is N^2. While it is difficult to guess the value of the prefactor α_c, the maximum number of stored bits must be proportional to the number of synapses. Many efforts have been made to improve upon this storage capacity, however, for a fully connected network it is possible to prove that the maximum number of stored independent bits cannot be greater than $2N^2$, or $2N$ uncorrelated patterns. More generally, if all neurons are not connected to each other, then the maximum number of patterns that can be stored is limited by the average number of synapses per neuron.

The loss of memory on reducing the number of synapses occurs when the synapses are set to zero a priori, independent of the information to be stored. This is a clue to the motivation for subdivision. The storage capacity would not be reduced if the synapses are set to zero because a zero value is appropriate to the information that is to be stored.

There may be reasons that are quite independent of storage considerations that the brain does not make use of a fully connected network. The most well known of these is the "connection problem." Three-dimensional space does not allow us to connect all neurons because of the difficulties in packing all of the connections into a volume in the presence of communication delays and heat-dissipation constraints. This problem is familiar to those who study the problems of designing massively parallel computers. While the connection problem might explain why the brain is not fully connected, it does not reveal the reason for nonuniformity of function in the brain.

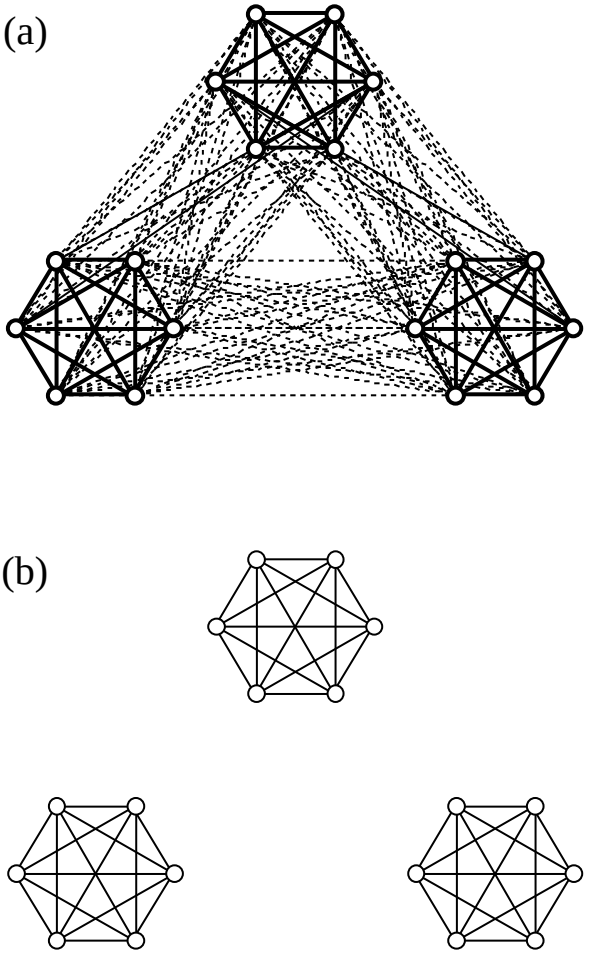

Figure 2.4.2 Subdivided neural network formed out of three subnetworks. The subdivision may be achieved either by setting the synapses between the subnetworks to be systematically weaker than those within a subnetwork or by systematically reducing the number of synapses between subnetworks. The former is indicated by the relative thickness of the lines in (a). In (b) the extreme case of a completely subdivided network is shown where the synapses between the subdivisions have been removed. ∎

We summarize our fundamental question as follows. For a network where each neuron is connected to every other neuron, the number of imprints that can be recalled, αN, is proportional to the number of neurons N with a constant of proportionality $\alpha < \alpha_c$ somewhat dependent on the particular imprinting rule and the desired properties of retrieval. When additional imprints are added, an overload catastrophe causes erasure of all information. Removing synapses or systematically

weakening synapses between subdivisions *inherently* results in a decreased storage capacity. Why then subdivide the brain?

2.4.1 *The left-right universe*

To begin to answer this question, consider first an artificial world composed of pictures with independent (uncorrelated) left and right halves (Fig. 2.4.3). This means that any left half that is seen in the universe may be found with any right half. A concrete example is the set of possible first and last name initials, where all letters of the alphabet might appear on the left, as they might on the right. Our task is to design a neural network for an organism in this artificial world. We assume that the pictures are mapped directly onto the network so that they are represented point by point as the neuron activity pattern. We will consider this example in detail, since it captures many of the essential concepts that will be relevant later.

As we have discussed, a fully connected network of N neurons is capable of recalling αN distinct and uncorrelated pictures. We can represent the stored pictures using the notation (L_i,R_i), where R_i is the right half of the ith picture and L_i is the left half of the ith picture. Then the stored images are of this form with i in the range $\{1,...,\alpha N\}$. In order for the pictures not to be correlated, the left sides of all stored pictures must be different from each other, as must be the right sides. If the pictures that the organism encountered in the universe were indeed distinct and uncorrelated, this is the best that can be expected from Hebbian training. However, in the left-right universe, the pictures that might be encountered *are* correlated.

Let us divide the network into left and right hemispheres by cutting all of the synapses running between them. The left hemisphere receives the left part of each picture and the right hemisphere receives the right part of each picture. Each of the hemispheres has $N/2$ neurons. Using Hebbian imprinting, each hemisphere can store $(\alpha N/2)$ distinct half-pictures. Because the subnetworks are half as large as the full network, the number of patterns that can be stored is half as many and each pattern is also half as large. The storage of the left hemisphere is of left halves of pictures, L_i. The storage of the right hemisphere is of right halves of pictures, R_i. In both cases i takes values in the range $\{1,...,\alpha N/2\}$. Storage in the left hemisphere is independent of the storage of the right hemisphere. When we test for recall, each of the patterns stored by the left hemisphere can be combined with each of the patterns in the right hemisphere to obtain a different stored picture. These are composites of the imprinted pictures. Thus the subdivided network stores a total of $(\alpha N/2)^2$ composite pictures of the form (L_i,R_j), where both i and j are taken independently from the range $\{1,...,\alpha N/2\}$. Each of these $(\alpha N/2)^2$ pictures may be encountered in the left-right universe. Since the number of neurons N is large, $(\alpha N/2)^2$ is much larger than αN. Cutting the synapses between the hemispheres results in a huge increase in the number of pictures that can be stored in the network. For an organism in the artificial world, this is a significant advantage.

The retrieval process is different in the fully connected network and in the subdivided network. In the fully connected network, retrieval starts by presenting to the network an image that is close to one of the stored images (L_i,R_i). Somewhat over 50%

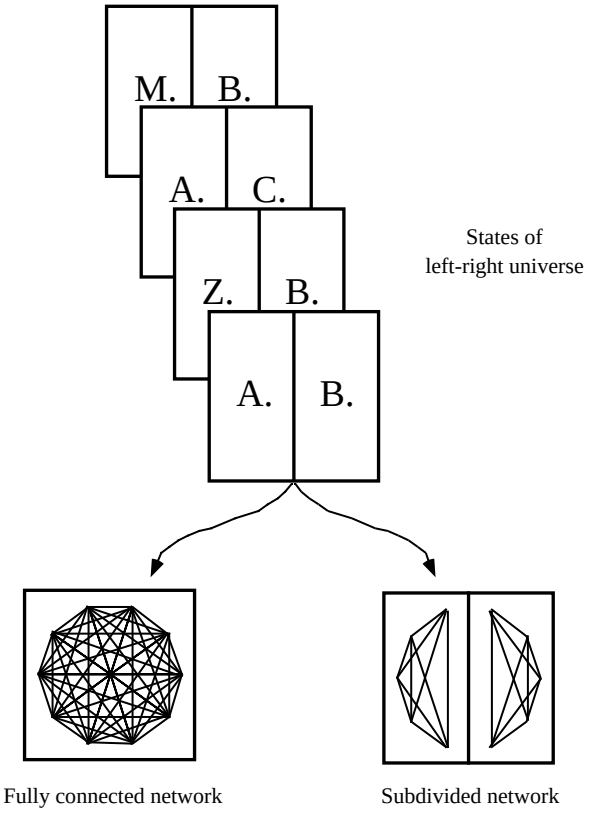

Figure 2.4.3 Illustration of the left-right universe, which consists of images that are composed out of independent left and right halves. The set of all initials is a simple example of such a universe. We could try to store these images on a fully connected network, or we could first subdivide the network into two hemispheres. The fully connected network would remember more completely independent images, but would fail to be able to store multiple images with the same left half or the same right half. The subdivided network would store independently the left and right half, and so would store many more images that would be composed of a selection of the stored left halves and the stored right halves. A biological analog of the left-right universe may arise in the control by the different brain hemispheres of left- and right-hand motion. ∎

of the neurons must be presented to the network in order to recover the full image. If we stored the patterns (L_1,R_1) and (L_2,R_2) and then the universe presents the pattern (L_1,R_2),the network will choose to settle into either (L_1,R_1) or (L_2,R_2), depending on which one of these is closer to (L_1,R_2). In either case we could say that the network is in error, but this error occurs for a state that the network never imprinted.

The subdivided network works on the retrieval of each half of the picture separately. It recognizes (L_1,R_2) from a pattern that is close to L_1 on the left and close to R_2

on the right. It might appear that there is a disadvantage because the subdivided network cannot use partial information from one half of the network to help in recall of the other half. However, it is important to recognize that this is actually dictated by the nature of the information in the left-right universe and only incidentally by the subdivision of the network. Using information from the left half to help on the right would only lead to errors, since it was assumed that there is no correlation between the two parts of the information.

It is significant that the training of the subdivided network was achieved with only $(\alpha N/2)$ imprints. The subdivided network recognizes many more pictures than were trained. In this way the network *generalized* from the training set to a much larger number of pictures. This works if we are able to select which pictures to train initially. We take $(\alpha N/2)$ of the pictures and make sure that all of the left halves and all of the right halves are different. We imprint these pictures. From the perspective of the storage of complete patterns, what we have done may appear quite strange. It is true that we have caused the network to store $(\alpha N/2)^2$ patterns, but aren't all of these really only a few patterns? Yes, they are all related to the imprinted $(\alpha N/2)$ patterns. The point is that in the artificial world, where the left and right parts of the image are independent, we *want* to store the $(\alpha N/2)^2$ different combinations rather than only a particular set of complete patterns.

Let us consider an alternative design of an organism in the left-right universe—a different way of subdividing the network. Instead of cutting the synapses between left and right hemispheres, we cut the synapses between the top and bottom halves of the network. In this case each half of the network acts to store $(\alpha N/2)$ half pictures. The top half stores the top part of each picture. The bottom half stores the bottom part of each picture. Now we cannot claim that the network stores $(\alpha N/2)^2$ pictures, because different combinations of top and bottom are not possible pictures in the universe. Instead the network stores at most only a total of $(\alpha N/2)$ of the possible pictures. There is an additional problem in retrieval, because information from the top cannot be used to help with retrieval of the bottom part of the image, and information from the bottom cannot be used to help with the top. In order to retrieve a particular image, we must have over 50% of the neurons correct in the top half and over 50% of the neurons correct in the bottom half. We have degraded the network storage with no compensating advantages. We have also created a whole host of undesirable memories that are not real. These undesirable memories are combinations of stored top and bottom halves of pictures.

From this discussion we learn that subdividing a network can improve dramatically the storage of patterns. However, the effectiveness of subdivision requires direct matching to the nature of the information:if we know that the organism lives in the left-right universe we can cut synapses between left and right hemispheres.

2.4.2 *Imprinting correlated patterns*

The advantage of subdivision in terms of the number of pictures that can be stored is not the whole story for the left-right universe. The fully connected network actually fails when patterns that are imprinted are significantly correlated. We have,until now,

considered only uncorrelated pattern storage in the fully connected network. In the left-right universe, the independence of information between the two halves of the pictures means that the patterns themselves are correlated. Two or more patterns that are to be remembered may have the same right halves and different left halves. When we imprint such correlated information in a fully connected network using Hebbian imprinting, the patterns are not stored. Qualitatively, the problem arises because the right-hand side of the network does not know which of the left sides to reconstruct. When we try to retrieve one of the stored patterns, the result on the left is retrieval of some intermediate picture that is neither of the desired memories. The degree of failure of the network depends on the number of pictures that are imprinted with the same right halves. Memory degradation occurs for just two imprinted pictures. Failure becomes explicit when there are as few as three imprinted pictures. It is, however, simplest to consider first the case of four imprinted patterns, all of which have the same right halves.

We can see the failure of a fully connected network analytically by considering the update of neurons when starting from one of the imprinted patterns. Extending Eq. (2.2.17) to the case of four patterns we have:

$$\begin{aligned} s_i(1) &= \text{sign}\Big(\sum_j J_{ij} s_j(0)\Big) \\ &= \text{sign}\Big(\xi_i^1 \sum_{j\neq i} \xi_j^1 s_j(0) + \xi_i^2 \sum_{j\neq i} \xi_j^2 s_j(0) + \xi_i^3 \sum_{j\neq i} \xi_j^3 s_j(0) + \xi_i^4 \sum_{j\neq i} \xi_j^4 s_j(0)\Big) \end{aligned} \tag{2.4.1}$$

We assume that we are looking at the value of a neuron in the left half of the network $i \in \{1,...,N/2\}$ and the four patterns $\xi_j^1, \xi_j^2, \xi_j^3, \xi_j^4$, are identical in the right half of the network $j \in \{N/2,...,N\}$. We split the sums in Eq. (2.4.1) into separate sums over the left and right halves of the network so that:

$$\begin{aligned} s_i(1) = \text{sign}\Big(&\xi_i^1 \sum_{j\neq i}^{N/2} \xi_j^1 s_j(0) + \xi_i^2 \sum_{j\neq i}^{N/2} \xi_j^2 s_j(0) + \xi_i^3 \sum_{j\neq i}^{N/2} \xi_j^3 s_j(0) + \xi_i^4 \sum_{j\neq i}^{N/2} \xi_j^4 s_j(0) \\ &+ (\xi_i^1 + \xi_i^2 + \xi_i^3 + \xi_i^4) \sum_{j=N/2+1}^{N} \xi_j^1 s_j(0)\Big) \end{aligned} \tag{2.4.2}$$

We have collected the sums over the right halves together since they are the same. We can test the stability of the first imprinted pattern. Setting $s_i(0) = \xi_i^1$, the first sum and the last sum are just $N/2$

$$s_i(1) = \text{sign}\Big(\xi_i^2 \sum_{j\neq i}^{N/2} \xi_j^2 \xi_j^1 + \xi_i^3 \sum_{j\neq i}^{N/2} \xi_j^3 \xi_j^1 + \xi_i^4 \sum_{j\neq i}^{N/2} \xi_j^4 \xi_j^1 + (2\xi_i^1 + \xi_i^2 + \xi_i^3 + \xi_i^4) N/2\Big) \tag{2.4.3}$$

The remaining three sums are random walks, because the left sides of the patterns are assumed to be uncorrelated. They have a typical size of $\sqrt{N}$ and are smaller than the last set of terms. So we can ignore the first three terms. As we look at different values of $i \in \{1,...,N/2\}$, one-eighth of the time we will have pattern 2,3,4 opposite the first pattern $\xi_i^1 = -\xi_i^2 = -\xi_i^3 = -\xi_i^4$. In this case the neuron will flip after the first update. If

one-eighth of a pattern changes after a single update, it is not stable. Since the same argument holds for each of the imprinted patterns, they were not stored.

Let us now think about the case of three patterns imprinted with the same left halves. Instead of Eq. (2.4.3) we have:

$$s_i(1) = \text{sign}(\xi_i^2 \sum_{j \neq i}^{N/2} \xi_j^2 \xi_j^1 + \xi_i^3 \sum_{j \neq i}^{N/2} \xi_j^3 \xi_j^1 + (2\xi_i^1 + \xi_i^2 + \xi_i^3) N/2) \tag{2.4.4}$$

This implies that whenever the other two patterns differ from the first at a particular neuron $\xi_i^1 = -\xi_i^2 = -\xi_i^3$, which happens 25% of the time, then the result is dependent upon the overlap of the three states. Specifically, in this case we have

$$s_i(1) = -\xi_i^1 \text{sign}(\sum_{j \neq i}^{N/2} \xi_j^2 \xi_j^1 + \sum_{j \neq i}^{N/2} \xi_j^3 \xi_j^1) \tag{2.4.5}$$

The argument of the sign function would always have to be negative in order for the imprinted pattern to be recovered. Statistically, the sign function will be negative or positive with equal probability. When it is negative,the whole pattern is stable. When it is positive,25% of the initial pattern will not be recovered after a single iteration and the pattern is unstable. Thus for three imprinted patterns, on average half of the patterns will be stable. The stability of these patterns is based upon the sign of the random-walk terms in Eq.(2.4.5). Imprinting any other pattern, correlated or uncorrelated, on the network will destabilize them.

In this section we have shown that the fully connected network fails for correlated patterns.Earlier, in Section 2.4.1, we discussed storage of uncorrelated pictures in the same network. If we have control over the order of pictures that are presented to the network, we can choose uncorrelated pictures to imprint. However, in the left-right universe, we should allow for an arbitrary order of the possible pictures. Some of the pictures will have the same left or right halves. In this case, the fully connected network with Hebbian imprinting will fail. Thus it is necessary to modify the fully connected network to work in the artificial left-right universe,and subdivision of the network is one way to do this. A network without synapses between left and right hemispheres does not suffer from this failure.

The real world is not constructed out of independent left and right pictures, at least the visual field is not. Can we make any sense of the actual subdivision of the brain (specifically the cerebrum) into left and right hemispheres from this model left-right universe? We can, at least in part, by recognizing that both tactile sensation and motor control of the arms and legs requires states that are left-right independent. Motor control requires neural activity patterns that describe (or prescribe) the motion. If we were to try to store the possible patterns of motion of the two hands in a uniform network,the actions of one hand would always be directly related to the actions of the other hand. If we want to be able to do one of several actions with the left hand for the same action of the right hand,then subdividing the network that stores the pattern of neural activity makes sense,and may even be necessary. Of course we would like there to be coordination between actions of the two hands or legs. This

means that we need a balance between the independence and dependence of the patterns in the two different divisions. We will investigate partially subdivided networks as one way to achieve this balance.

2.4.3 *Separating independent information*

The example of left-right motor control is a special case in which the simple model of subdivision might help, and might even correspond to an important example of subdivision of the brain. We can generalize this example by recognizing that the information we process is highly correlated. One of the reasons for the correlation is that different aspects of the information are independent of each other in similar manner to the independence of the left-right universe. It is this independence that gives rise to correlations.

We therefore recognize that there are two tasks.First, we must in some way identify which parts of the information to be stored are independent (uncorrelated) and separate these parts of the information. Then we must store these different types of information in different parts of the network. Achieving this will enable tremendous increase in storage of the correlated patterns.Fig. 2.4.4 shows a simple model for the function of sensory processing consistent with this concept. Sensory information is separated by input processors to distinct channels. The input processors are presumed to be composed of feedforward neural networks that are illustrated only schematically. The information is then imposed on a subdivided attractor network that serves as a content-addressable memory.

If the information in the different channels were completely independent, the channels should be completely independent and the entire problem would be to identify what the channels should be. However, the information is not usually completely independent. This suggests that we adopt a model where the network is partially subdivided, with weaker or fewer synapses between the subdivisions of the attractor network. This is the model that we will adopt and investigate. Before pursuing this approach we discuss two more examples of the relevance of this architecture to human information-processing: vision and language.

2.4.4 *Sensory processing: color, shape and motion in vision*

The human visual system does not take advantage of the two hemispheres of the brain to divide the visual information right from left because the left and right parts of the visual field are not independent. There is a large interconnection area called the corpus callosum that connects the visual areas in the two hemispheres. Instead, detailed mapping of the visual cortex has revealed that visual processing separates three attributes of the information: color, shape and motion. The implication of the separate processing of these three attributes using, in effect, a preprocessing step to separate them, is that these information categories are partially independent. For example, visual fields with different shapes can have the same colors. Or, vice versa, the same shapes can have different colors. This independence has been used in the design of the genetically encoded structure of the initial visual information-processing.

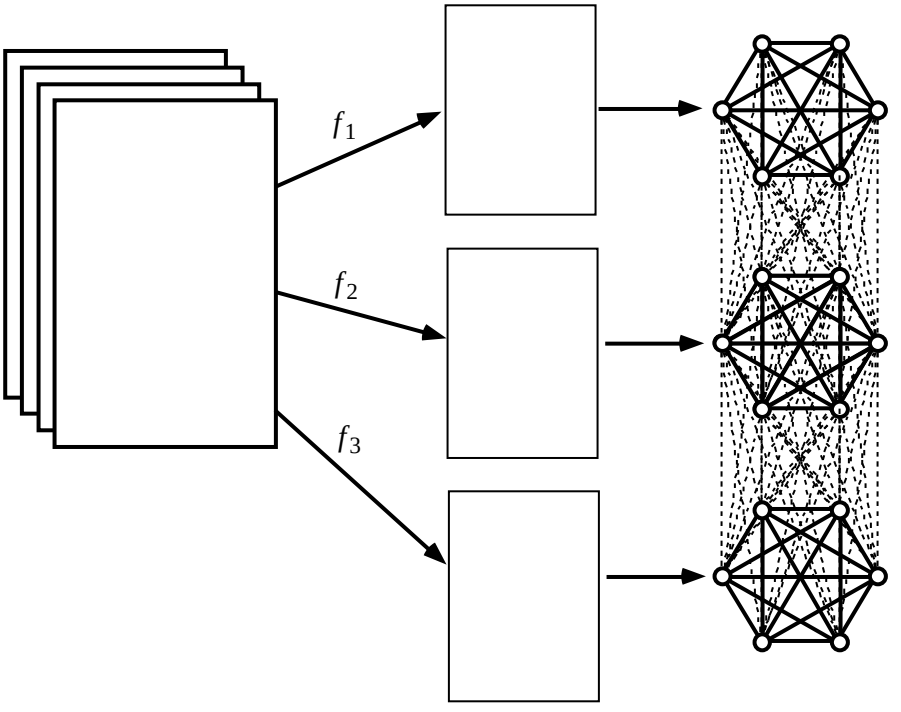

Figure 2.4.4 Schematic illustration of a model for sensory processing that first separates the information into distinct channels, each of which corresponds to a different attribute of the input. The separated attributes are then imprinted on distinct subdivisions of a neural network. This approach is effective if the different attributes of the information are independent, or at least partially independent. ∎

The use of these three channels in the visual system can be recognized in our use of attributes to identify objects. In describing objects, we generally distinguish distinct types of attributes—color, shape and action/motion. Within each of these attribute categories we can construct a list of attributes such as

color: RED, GREEN, BLUE, ORANGE, PURPLE, WHITE, BLACK, ...

shape: ROUND, OVAL, SQUARE, FLAT, TALL, ...

action/motion: STATIONARY, MOVING-LEFT, MOVING-RIGHT, RISING, FALLING, GROWING, SHRINKING, ...

The existence of three attribute categories enables a large number of descriptive categories to be constructed. A description is composed out of a selection of one attribute from each category. The number of descriptive categories is the product of the numbers of attributes of each type.

By subdividing a network into three subnetworks and separating the color information to one subnetwork, the shape information to the second,and the movement information to the third,it is possible for the network to identify categories such as: RED ROUND MOVING-LEFT, RED ROUND FALLING, BLUE SQUARE MOVING-LEFT, and BLUE ROUND FALLING. The network receiving color information identifies the color, and so on. In a fully connected network these categories would each require separate identification (and correlations would actually cause the network to fail). As with our descriptions,in the subdivided network the total number of categories is the product of the number of categories stored in each subnetwork.

We caution that the shape, color and motion attributes of the information are not completely independent, and neither is their processing in the brain. Partial subdivision implies correlations between the different attributes are also significant. Particularly in the natural world,there are important correlations between the overall shape of an object, its color, and both its direction and likelihood of motion. For example, leaves have a set of characteristic shapes and they are usually green. Tree trunks and their associated vertical or branching shapes are usually brown. The ways in which leaves are likely to move are not the same as the way tree trunks are likely to move. If we used a completely subdivided network for vision,after imprinting brown stationary trees and green rustling leaves we would also remember brown rustling tree trunks and green stationary tree trunks. In order not to lose the color-shape-motion relationships, we must be able to store the correlations between these different attributes. This may be done in a partially subdivided network using the weaker or fewer synapses that run between the different subdivisions.

2.4.5 *Language and grammar: nouns, verbs and adjectives*

If subdivision provides advantages for neural network function, then this should be particularly manifest in man-made constructs. These constructs are likely to reflect the architecture of the brain and therefore mirror the use of subdivision. In the context of vision, one might consider the use of color and shape in abstract art,as well as the decoration of man-made objects (e.g.,package labels).Studies of these constructions might help develop an understanding of the human visual system. Another source of information is human language. Known as "natural language" in the artificial intelligence community, to contrast it with computer languages, the spoken or written language is a man-made construct that has been studied for many years by linguists as a source of information or insight into the functioning of the human brain.

Linguists differentiate between the grammatical and semantic aspects of language construction. Loosely speaking, grammar is the structure of well-formed sentences, while semantics is the content of the sentences. It is grammar, which is much more amenable to formal studies, that has been considered to reflect the architecture of the brain. A basic premise in the field of modern linguistics is that common features in the grammar of different languages exist and are the primary clue to the inherent brain architecture. The most recent widely accepted linguistic theory is the transformational grammar. It suggests that there exists an underlying representation that is transformed upon output into the usual grammatical form of sentences.

Sentences are interpreted upon input to reconstruct the underlying representation. This resembles our model of the neural architecture, with feedforward input processing that leads to the subdivided attractor networks and, we add here, output processing from the subdivided network to the motor controls. The input and output mappings that form the grammatical constructions used in a particular language are not completely universal and must be trained. This will not be the focus of our attention here.

Our objective at this stage in describing the connection between grammar and our model is quite modest: to make contact with one of the most fundamental aspects of grammatical construction that is familiar to everybody—the existence of parts of speech in sentence construction. Indeed, without the existence of parts of speech, there would be no meaning to the term "grammar." Grammar investigates the construction of sentences out of words. Words are separated into categories that are the "parts of speech," such as nouns, verbs, adjectives, and adverbs. The central role of grammar is to describe the rules by which properly formed sentences are constructed out of the parts of speech.

In order for words to be stored in the brain, some appropriate representation must exist in terms of neuron activity patterns. We do not know what this representation is, nor how universal the representation is. However, assuming some representation, we can ask how the organization of words into parts of speech can be realized in the brain. One way is to attach a label to each word that indicates what part of speech it is, and to store each word with its label as a pattern in a uniformly connected neural network. When we use a particular word, the label can serve to identify how it should be used in a sentence. This is how dictionaries are organized. After each word appears the usage—part of speech (abbreviated n, v, adj, adv, etc.)—identifying how it may be used in a sentence. There are some technical problems with storing patterns which incorporate labels in this way. Since the same label (part of speech) applies to many words, this will not work in a conventional attractor network. There are ways to overcome this problem, but we will not take this route here.

Instead we describe an alternative that makes use of network subdivisions. The architecture is similar to the model for vision that was used in the previous section, or the more general model of Fig. 2.4.4. We simplify the construction by considering only three parts of speech—nouns, verbs and adjectives. We assign each part of speech to a particular brain subdivision and assume that visual (reading) or auditory processing separates the information stream into three parts. The separation of the information is equivalent to parsing sentences using the grammatical sentence structure, a process that is reasonably well-understood. The importance of input processing provides a reason for the need for consistency in grammatical construction. After the initial processing parses the sentence, the parts (noun, verb and adjective) are transferred to distinct brain subdivisions. In order to generate sentences for writing or speaking, an output processor (presumably another set of feedforward networks) is necessary to take the content of the brain subdivisions (noun, verb and adjective) and compose a sentence. This output processor precedes the motor control of speaking, writing or typing. It reimposes the grammatical construction of sentences.

In this picture the brain as a whole does not store words, per se, but rather phrases or short sentences that consist of one each of a noun, verb and adjective (n, v, adj). If we wanted to discuss how this model is capable of continuous language we would add a time evolution of the network that causes a transition from one word triple (n, v, adj) to the next. We limit our consideration of this model to an understanding of the role of subdivision by comparing a model with completely separated subnetworks with another model that stores phrases or short sentences in a fully connected network.

The comparison is illustrated in Fig. 2.4.5. We take two networks with the same number of neurons—a uniformly connected network and a network divided into three parts. Since the independent pattern storage capacity grows linearly with the number of neurons, the number of short sentences that can be stored in the uniform network is three times the number of patterns that can be stored in each of the three pieces of the subdivided network. For this example we take quite small networks, so that each of the subdivisions can store three words coded appropriately. The uniform network would then be able to store nine sentences with three words each. We choose to imprint the nine sentences that are shown in Fig. 2.4.6 on the left. On the subdivided network we can only imprint three sentences. However, twenty-seven composite sentences would be recognized.

The difference between the set of sentences that can be remembered by the full network and the set that can be "remembered" by the subdivided network are related to the distinction between the grammatical and semantic content of a sentence. The complete network knows more full sentences, but does not have knowledge of the divisibility of the sentences into parts that can be put together in different ways. It does not even recognize the existence of word boundaries. The subdivided network knows the parts but does not know the relationship between them, thus it knows grammar and it knows the individual words, but it does not know the semantic content. For example, it does not know who it is that fell. The subdivided network generalizes from the three imprinted sentences to twenty-seven sentences. This generalization is based on the grammatical construction of the sentence. The fully connected network does not generalize in this way because it remembers the specific imprinted sentences to the exclusion of all others.

The field of linguistics as well as our intuition suggests that the actual process in the human brain lies somewhere between these extremes. Sentences make sense or are "grammatically correct" if properly put together out of largely interchangeable parts. However, a recalled event is described by a specific combination. Language, whether written or spoken, is generated by each individual out of sentences. The particular sentences that are used were not necessarily learned. Whether read or heard, language is understood by each individual by recognizing the component words. Yet much of the new meaning that is learned is contained in the interrelationship of words. A specific combination of words can be remembered by an individual and repeated. However, in general such memorization is not easy and is not as permanent as the memory of individual words.

It is possible to achieve an intermediate balance between storage of components and complete imprints by use of a partial interconnection between subnetworks. We

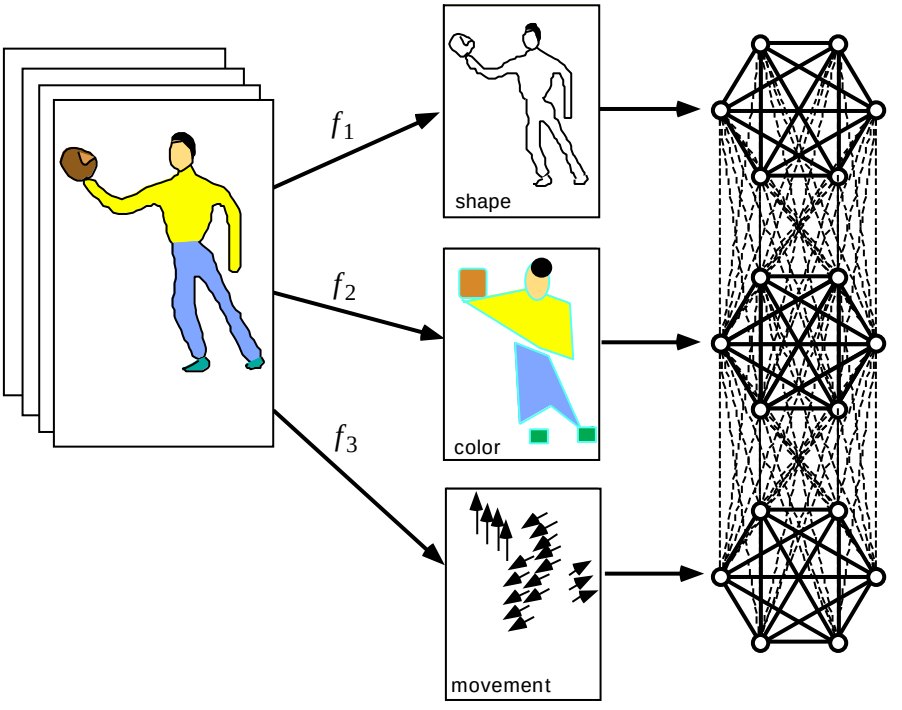

Figure 2.4.5 A practical example of the separation of processing into separate channels is the separation of visual processing into color, shape and motion. A significant body of experimental literature indicates that visual processing is separated into different channels. The channels are not fully characterized but are roughly considered to correspond to color, shape and motion. ∎

will investigate a few properties of partially subdivided networks in Section 2.5. One of the features of the partially subdivided network is that the synapses that run between subnetworks impose "compatibility" relations between the patterns stored in each subdivision. Some combinations of subpatterns are stable while others are not.

The subdivided network provides a systematic method for information organization in terms of elements (the stable states of subnetworks) which are organized in element-categories (the stable states of a particular subnetwork) and the compatibility relationships between elements as dictated by the inter-subnetwork synapses. This is indeed reminiscent of the structure of grammar, where nouns, verbs and adjectives and other parts of speech are categories that have elements,and there are compatibility relations among them. It is tempting to speculate that different subdivisions of the brain are responsible for the classification of words into parts of speech,and that the ability to combine them in different ways results from balancing the strength of inter-subnetwork synapses and the intra-subnetwork synapses which store representations

Fully connected network			Subdivided network					
Imprinting and Retreival			Imprinting			Retreival		
Big	Bob	ran.	Big	Bob	ran.	**Big**	**Bob**	**ran.**
Kind	John	ate.	Kind	John	ate.	Big	Bob	ate.
Tall	Susan	fell.	Tall	Susan	fell.	Big	Bob	fell.
Bad	Sam	sat.				Big	John	ran.
Sad	Pat	went.				Big	John	ate.
Small	Tom	jumped.				Big	John	fell.
Happy	Nate	gave.				Big	Susan	ran.
Mad	Dave	took.				Big	Susan	ate.
Shy	Cathy	slept.				Big	Susan	fell.
						Kind	Bob	ran.
						Kind	Bob	ate.
						Kind	Bob	fell.
						Kind	John	ran.
						Kind	**John**	**ate.**
						Kind	John	fell.
						Kind	Susan	ran.
						Kind	Susan	ate.
						Kind	Susan	fell.
						Tall	Bob	ran.
						Tall	Bob	ate.
						Tall	Bob	fell.
						Tall	John	ran.
						Tall	John	ate.
						Tall	John	fell.
						Tall	Susan	ran.
						Tall	Susan	ate.
						Tall	**Susan**	**fell.**

Figure 2.4.6 Illustration of the use of subdivided networks in the context of language. A fully connected network with enough neurons to store exactly nine sentences shown on the left may be imprinted with and recognize these sentences. If the network is divided into three parts it may be imprinted with only three sentences (center). However, because each subnetwork functions independently, all twenty-seven sentences (right) that are formed as composites of the imprinted sentences are recognized. Comparing left and right columns suggests the difference between semantics and grammar in sentence construction. ∎

of each word. There is even some biological evidence for the separation of nouns and verbs in different parts of the brain. We could take a step further and consider the relationship of the subdivisions of the brain that store noun and verb representations with other parts of the brain. For example,it makes sense to speculate that the subdivision that stores nouns would be more strongly connected by synapses to sensory-

processing parts of the brain as compared to motor-processing parts. In contrast, verbs would be likely to be more strongly connected to motor control than most of the sensory processing (but also to the motion-detection subdivision of the visual system). This might even be part of an explanation of why words are divided into the categories of noun and verb, rather than some other categorization.

The discussion of the previous paragraph is the beginning of an approach to discussion of the architecture of the brain based on an understanding of how subdivided neural networks function. A more detailed discussion of how this approach might help in developing an understanding of neurophysiology will be given in Chapter 3. However, we mention here the implication that one might use the logic of grammar to represent more generally the function of the brain. To do this we would expand the articulated sentence to include additional "unvoiced" words in new categories representing the state of brain subdivisions other than the language-related ones.

2.5 Analysis and Simulations of Subdivided Networks

Our objective is to consider the advantages of subdivided networks in the context of sensory processing or, more generally, in the context of pattern-recognition tasks. The advantage of subdivision arises when the information is naturally subdivided so that combinations of imprinted subnetwork states also represent desirable states to be recalled by the network. We call these combinations of subnetwork states composite states. For completely subdivided networks, the analysis is immediate (Section 2.5.1). For partially subdivided networks, the analysis is discussed in Sections 2.5.2-2.5.4. Partially subdivided networks are relevant to pattern recognition when the information may be divided into partially but not completely independent parts. Thus a determination of the interdependence of recalled subnetwork states is relevant. It is assumed that for a particular pattern-recognition task, a balance is desirable between independence and correlation of subnetwork states. The central question is whether it is possible to achieve an adjustable intermediate balance between storage of complete neural patterns and storage of composite states.

We will use partially subdivided networks consisting of a conventional network of N neurons with Hebbian imprinting, where the strength of synapses between q subdivisions of $N' = N/q$ neurons are reduced by a factor g compared to the synapses between neurons within each subnetwork. We can expect that dilution of inter-subnetwork synapses, with g the fraction of remaining synapses, will lead to similar results (Question 2.5.1 on p. 364). It is important to distinguish the subdivided network from a randomly diluted network. Random dilution would sever synapses selected at random. Dilution of inter-subnetwork synapses results in storage of composite patterns. This would not occur for random dilution.

Consider a network with predefined subdivisions. The training of the network is performed by imprinting complete neural states. Since subdivision is favorable only when it is desirable to store and recognize composite patterns, we measure the stability of various composite patterns such that the state of each subnetwork corresponds

to its state for one of the imprinted patterns.* Thus,two questions to be asked about the capacity of the network are: (1) How many complete neural states can be stored in the network? and (2) How many composite patterns are stored in the network?

2.5.1 *Completely subdivided networks*

For either fully connected or completely subdivided networks, the capacity is obtained from the results of Section 2.3 on fully connected networks. The maximum capacity of a large attractor network of N neurons for storage of complete neural states is $\alpha_c N$, where $\alpha_c = 0.145$ for Hebbian learning when a small fraction of errors in retrieval is allowed. For a network of N neurons completely subdivided into q subdivisions,the maximum capacity f or complete neural states is $\alpha_c N/q$, which is lower than for the undivided network. However, since storing $\alpha_c N/q$ states results in all possible combinations of these substates being stored,the number of composite states stored is $(\alpha_c N/q)^q$, which is much larger than $\alpha_c N$ for large N and not too large q. This result follows directly from the linear dependence of the network capacity on the number of neurons.

As an example, for a network of 100 neurons, the storage capacity is 14.5 states for the full network. When subdivided into two halves, the network capacity is 7.25 full memories,and $(7.25)^2 - 7.25 = 46$ composite patterns in addition to the full memories. When subdivided into four subnetworks,the same 100 neurons store 3.75 complete states and 195 composite patterns. This example is described and simulated more fully below (see Fig. 2.5.1 through Fig. 2.5.4).

As an exercise we might calculate the number of subdivisions that results in storage of the largest number of composite patterns:

$$\frac{d}{dq}\left(\frac{\alpha_c N}{q}\right)^q\Bigg|_{q=q_{opt}} = q\left(\frac{\alpha_c N}{q}\right)^q\left(\ln\frac{\alpha_c N}{q} - 1\right)\Bigg|_{q=q_{opt}} = 0$$

$$q_{opt} = \frac{\alpha_c}{e} N \tag{2.5.1}$$

For this number of subdivisions, the number of neurons in a subdivision would be only $e/\alpha_c \approx 19$. The number of memories stored (assuming that the usual formula applies, which is only approximate in this small subnetwork limit) is $e \approx 2.7$, or less than 3 on average. The total number of independent memories stored is:

$$\left(\frac{\alpha N}{q_{opt}}\right)^{q_{opt}} = e^{q_{opt}} = e^{\alpha N/e} \tag{2.5.2}$$

* In the present discussion we neglect the inversion of imprinted substates. For example, for a network subdivided into two parts, we could consider a state formed out of the right half of one imprinted pattern and the left half of the same pattern inverted. Such states may also be stable.A bias in the relative number of ON and OFF neurons (see also Section 3.2.13),as is found in the brain, would lead such states to be less relevant.

# of Neurons	# of patterns in full network	# of patterns in 2 subdivisions	# of patterns in 3 subdivisions	q_{opt}	maximal # of patterns
100	1.45×10^{1}	5.26×10^{1}	1.13×10^{2}	5	1.48×10^{2}
1000	1.45×10^{2}	5.26×10^{3}	1.13×10^{5}	53	1.04×10^{23}
10000	1.45×10^{3}	5.26×10^{5}	1.13×10^{8}	533	3.01×10^{231}

Table 2.5.1 Table of the storage capacity for composite patterns in various subdivision schemes. If the number of composite patterns stored is maximized, the number of subdivisions q_{opt} and the number of memories stored is indicated. Note that the number of imprints needed to store a large number of composite patterns is not great. In particular it is only three for all cases in the last column.

This subdivision size is based on considering the maximum number of composite states which can be stored.

The main problem with this analysis is that it only applies if the information can be divided into independent parts consisting of N/q bits. We could also consider a more extreme case where all bits are independent. In this case there would be no need for synapses (or storage) since all 2^N patterns are possible. Nevertheless,the implication that an optimal subnetwork size only stores approximately e states may be of significance when we can adjust the pattern-recognition task to suit the capabilities of the neural architecture.

Table 2.5.1 indicates the potential advantage of subdivision if the information can be appropriately subdivided into aspects that can be mapped onto subdivisions of the network. The human brain (with 10^{11} neurons) has left and right hemispheres that are further subdivided into a hierarchy of subdivisions. Small divisions are sometimes modeled as having about 10^4 neurons in number. Further subdivisions into still smaller neuron groups may also occur in the brain.

2.5.2 *Summary of results on partially subdivided networks*

In Section 2.5.3 and Section 2.5.4 we analyze networks that are partially subdivided. In Section 2.5.3 simulations are used and in Section 2.5.4 a signal-to-noise analysis is used. Before proceeding, we summarize the results. All of the results on partially subdivided networks depend on the degree of subdivision. g sets the relative strength of inter-subnetwork synapses and intra-subnetwork synapses. For $g = 1$ we have a fully connected network,and for $g = 0$ we have a completely subdivided network. The simulations and signal-to-noise analysis show that

1. For $g = 1$ the maximal number of imprinted patterns may be stored and for $g = 0$ the minimal number of imprinted patterns may be stored with a continuous interpolation between them.
2. For $g = 1$ the lowest number of composite patterns may be stored and for $g = 0$ the largest number of composite patterns may be stored with a continuous interpolation between them.

3. For particular values of g a well-defined balance between complete patterns and composite pattern storage is achieved.

4. a. We should distinguish between composite patterns that have some subdivisions with the same imprinted pattern in them. For example, for three subdivisions,there are composite patterns with two of the subdivisions having the same imprinted pattern in them and the third subdivision with a different imprinted pattern. Patterns that have more than one subdivision with the same imprinted pattern continue to be stable to higher values of g. More specifically:

   b. The simulations study a network subdivided into four subdivisions.A composite pattern formed by setting two of the subdivisions to one imprinted pattern and two subdivisions to another is stable to higher values of g.

   c. The signal-to-noise analysis considers networks with q subdivisions. Let a be the smallest number of subdivisions occupied by an imprinted pattern in a particular composite pattern. Then for low storage,this pattern will be stable for all g satisfying

$$g < \frac{1}{q - 2a + 1} \tag{2.5.3}$$

   The significance of result 4 is that we can use the value of g to impose correlations between patterns stored in different subnetworks by stabilizing some composite patterns and not others.

5. When the number of subdivisions becomes large, Eq.(2.5.3) ceases to apply, and it becomes impossible to selectively impose correlations between patterns stored on different subnetworks. If we try to reduce g to allow composite states,then all composite states become possible. As a consequence, we learn that beyond a certain number of subdivisions, partial subdivision is essentially impossible. The network either behaves as a fully connected network or as a completely subdivided network. For many purposes it is thus undesirable to have more than a few subdivisions. The crossover point is calculated to occur for approximately seven subdivisions. This result has some significance for our understanding of the subdivision in the brain and brain function. For example,it is consistent with the 7 ± 2 rule of short-term memory. It is also of significance for our understanding of complex systems in general.

Another way to state result 5 is in the language of Section 1.3.6. A uniform network may be categorized as a complex material. Removing part of the network affects the smaller part but does not affect the larger part of the network. In contrast, for less than approximately seven subdivisions,at intermediate values of g, subdivided neural network function depends on each of its subdivisions. It is therefore in the category of complex organisms. For greater than seven subdivisions it can no longer be in the category of a complex organism. For large enough g it behaves as a fully connected network and is a complex material. For smaller g the network decouples and becomes

a set of independent networks. In this case it is unchanged under subdivision, like a thermodynamic system.

2.5.3 *Simulations of partially subdivided networks*

To evaluate the behavior of networks that are partially subdivided rather than completely subdivided,it is natural to perform simulations. These simulations,analogous to those of fully connected networks, test the stability of imprinted and composite neural states. In the following we use Hebbian imprinting and synchronous updating of an attractor network with 100 neurons partially subdivided into either two or four subdivisions. The imprinted patterns are chosen at random with an equal probability of the two neuron activities ±1. The procedure used for performing the simulations of subdivided networks is:

1. Generate p complete random neural states:

$$\xi_i^\mu = \pm 1 \qquad \mu = \{1,...,p\},\ i = \{1,...,N\} \tag{2.5.4}$$

2. Imprint these neural states on the synapses of the neural network:

$$J_{ij} = \begin{cases} \dfrac{1}{N}\displaystyle\sum_{\mu=1}^{p} \xi_i^\mu \xi_j^\mu & i \neq j \\ 0 & i = j \end{cases} \tag{2.5.5}$$

3. Write the matrix of synapses in block form corresponding to q equal size neuron subdivisions

$$J = \begin{pmatrix} J^1 & J^{1,2} & \cdots & J^{1,q} \\ J^{2,1} & J^2 & & J^{2,q} \\ \vdots & & \ddots & \vdots \\ J^{q,1} & J^{q,2} & \cdots & J^q \end{pmatrix} \tag{2.5.6}$$

where each superscripted J is an $N' \times N'$ matrix, $N' = N/q$. Diminish off diagonal blocks of synapses, which connect between q different subnetworks of equal size, by a factor g.

$$\begin{aligned} J^i &\leftarrow J^i & i &= \{1,...,q\} \\ J^{i,j} &\leftarrow g J^{i,j} & i,j &= \{1,...,q\}, i \neq j \end{aligned} \tag{2.5.7}$$

4. Find the number of imprinted and composite states that are stable under updating of the neurons.A composite state is composed from imprinted states in each subdivision. In general:

$$\begin{aligned} \zeta_i^\beta &= \xi_i^{\alpha_1} & \alpha_1 \quad & \{1,...,p\}, i = \{1,...,N'\} \\ \zeta_i^\beta &= \xi_i^{\alpha_2} & \alpha_2 \quad & \{1,...,p\}, i = \{N'+1,...,2N'\} \\ &\vdots & & \\ \zeta_i^\beta &= \xi_i^{\alpha_q} & \alpha_q \quad & \{1,...,p\}, i = \{(q-1)N'+1,...,N\} \end{aligned} \tag{2.5.8}$$

The number of stable composite states is:

$$p_{stable} = \sum_{\beta} \prod_{i} \delta\left(\zeta_i^{\beta}, \theta\left(\sum_{j} J_{i,j}\zeta_j^{\beta}\right)\right) \tag{2.5.9}$$

In the simulations, the number of stable memories for small p are counted by enumerating all combinations. For p greater than a few, the number of stable states is obtained by sampling. In both cases, averaging over many sets of imprinted states is performed. Note that no errors are allowed in recall.

Fig. 2.5.1 shows the results of simulations of a network with 100 total neurons and two subdivisions. It shows separately the number of stable composite states that were not imprinted (Fig. 2.5.1(a)),and the imprinted states (Fig. 2.5.1(b)).Fig. 2.5.2 shows the total number of stable states. Different curves show the result of diminishing the inter-subnetwork synapses by g ranging from 0 to 1 in increments of 0.1. The maximum number of stable imprinted states is for a network that has not been subdivided, $g = 1$. The maximum is obtained for 13 or 14 imprints and is about 11 memories. This is the same as the earlier Fig. 2.2.6. The maximum number of composite states recalled is for the completely subdivided network. The maximum is obtained for approximately 7 imprints, resulting in recall of 45 composite states. These numbers approximate the expected results given by the analytic treatment. Note that the analytic treatment need not give exactly the same result as the simulation, because it assumes that N, N' are very large, and it allows some error in the network recall.

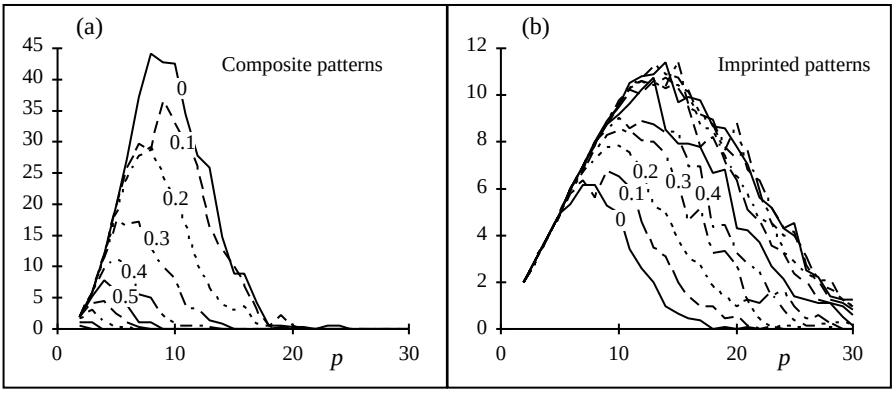

Figure 2.5.1 The number of stable memories after imprinting p patterns (horizontal axis) on a subdivided neural network with 100 total neurons and two subdivisions. (a) shows the number of stable composite patterns composed of combinations of imprinted subnetwork patterns where the complete pattern is not an imprinted one. (b) shows the number of stable imprinted patterns. Note the difference in vertical scale. The different curves are labeled by the factor g which weakens the synapses connecting different subnetworks. The curves labeled 0 are for a completely dissociated network. ∎

Of particular interest in these simulations is the possibility of partially subdividing the network and achieving an intermediate balance between storing imprinted states and composite states. For interconnection strengths reduced by $g = 0.3$, and with 9 imprints of which nearly all are recalled,the number of additional composite states recalled is about 10. This example of the network subdivided into two parts illustrates the balance between complete and composite memories. However, the nature of stability of subnetwork combinations in a subdivided network is more effectively illustrated with additional subdivisions.

Imprinting on a network with four subdivisions results in various possibilities for composite patterns. As in step 4 (p. 349), letting $\xi_i^{\alpha_1}$ be the state of the ith neuron in the α_1 imprint, we can write the composite states using the notation $\zeta_i^{\beta} \rightarrow (\alpha_1, \alpha_2, \alpha_3, \alpha_4)$. The distinct types of vectors whose stability can be tested are distinguished by the equality or inequality of the α_i as shown in Table 2.5.2. The number of stable memories of each type for 100 neurons and after p imprints is plotted in Fig. 2.5.3, and totaled in Fig. 2.5.4.

For a completely subdivided network ($g = 0$) the storage capacity for imprints is just over 3, compared with the full network capacity of 11. The number of composite

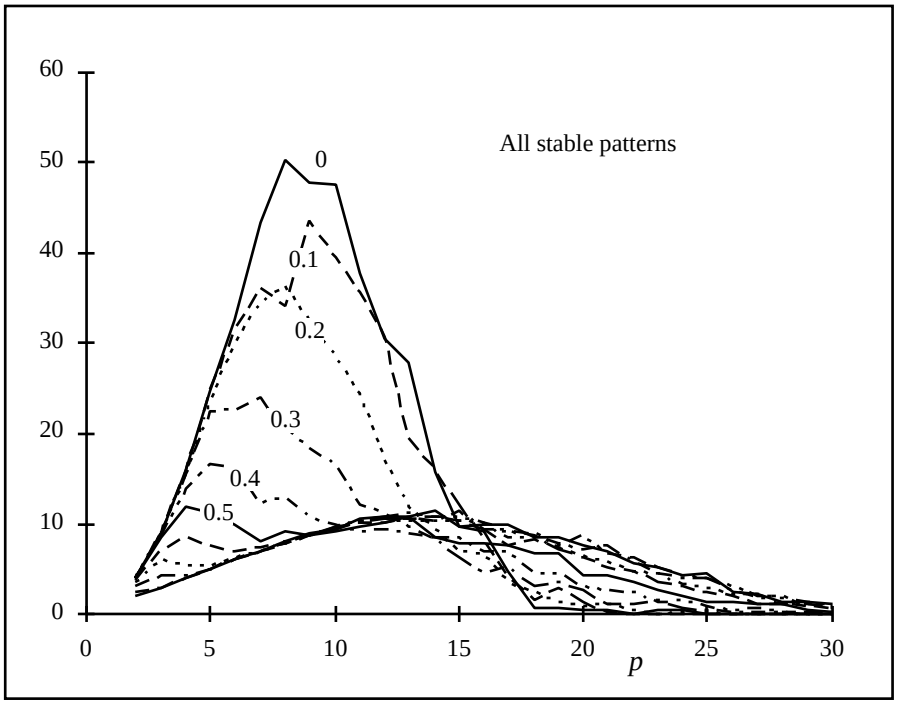

Figure 2.5.2 The total number of stable composite and imprinted states after imprinting p patterns on a subdivided neural network with 100 total neurons and two subdivisions. The value of g is indicated on each curve. ∎

Category	Number of Such States	Label	Schematic
$(\alpha_1,\alpha_1,\alpha_1,\alpha_1)$	p	Imprinted states	
$(\alpha_1,\alpha_1,\alpha_1,\alpha_2)$	$4p(p-1)$	Three equal substates	
$(\alpha_1,\alpha_1,\alpha_2,\alpha_2)$	$3p(p-1)$	Two & two equal substates	
$(\alpha_1,\alpha_1,\alpha_2,\alpha_3)$	$6p(p-1)(p-2)$	Two equal substates	
$(\alpha_1,\alpha_2,\alpha_3,\alpha_4)$	$p(p-1)(p-2)(p-3)$	Unequal neural substates	

Table 2.5.2 Different types of composite states for a network subdivided into four parts. The first type consists of substates that all originate from the same imprinted state—an imprinted state. The second type consists of substates of which three are from the same imprinted state and one originates from a different imprinted state. The other types are similarly defined. The number of states of each type is indicated (it is assumed that $p \geq 4$) in the second column. A label for each type, which is used in the figures and in the text, is given in the third column. A schematic is indicated in the last column. Note that the number of states in the last category is largest when p becomes large enough, however, for $p < 9$, the second to last category has a larger number of states. ∎

states recalled is nearly 400. When interconnection strengths are reduced by $g = 0.2$, it is possible to store between 6 and 7 complete memories while enabling the stability of 70 additional composite states at the same time. These composite states are roughly equally divided between those with two equal substates, and those with two & two equal substates. Other values for the interconnection strength can provide a distinct balance between the independence and dependence of subnetwork states.

Systematically, it is possible to see that composite patterns that have more than one subdivision containing the same imprinted pattern remain stable at higher values of g. When all substates arise from different imprinted states (Fig. 2.5.3(a)), the stability decreases very rapidly as g increases. The number of substate combinations with two equal substates (Fig. 2.5.3(b)) decreases almost as rapidly. In contrast, the stability of states with two & two equal substates (Fig. 2.5.3(c)), diminishes much more slowly. The number of states with three equal substates (Fig. 2.5.3(d)) is insignificant in these simulations. The greater stability of states with two & two equal substates at higher values of g is reasonable because the synapses between the subdivisions can contribute to the stability of each of the two parts of the composite pattern that arise from different imprints, even though the interactions between the two parts tend to destabilize each other. This will become more apparent through the analytic discussion in the following section.

2.5.4 *Signal-to-noise analysis of subdivided networks*

A signal-to-noise analysis of the stability of composite patterns in partially subdivided networks requires some care, because there are several different contributions to the signal and to the noise. Before we perform the analysis, it is helpful to discuss these

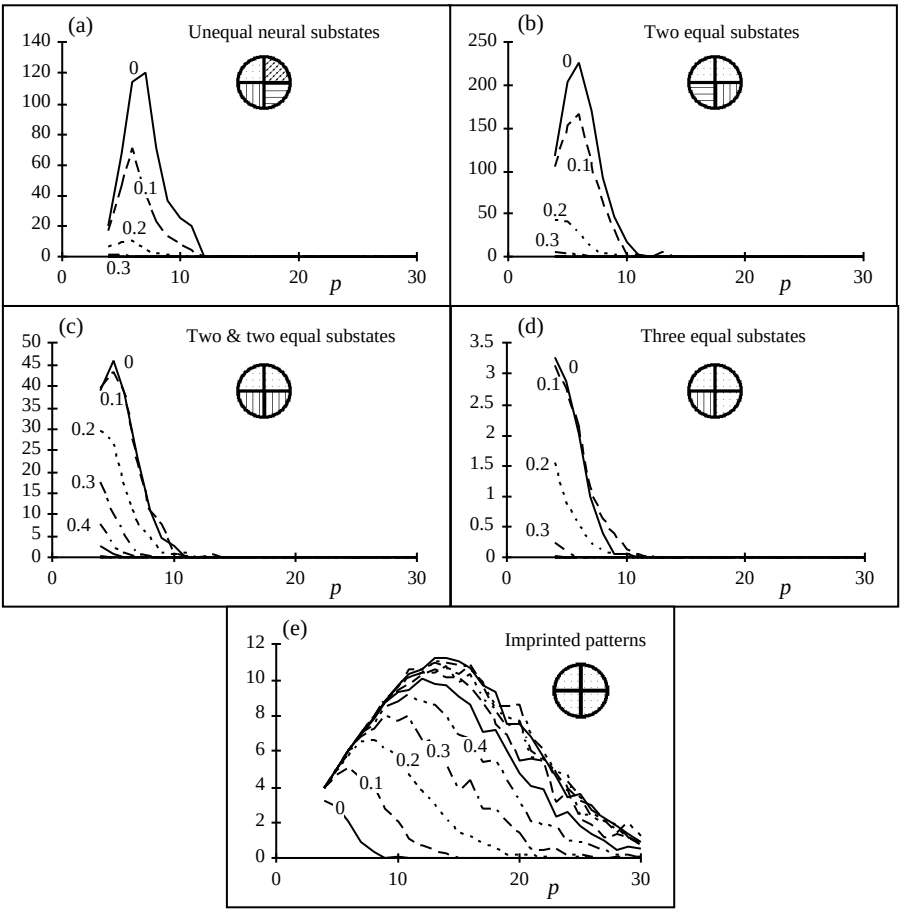

Figure 2.5.3 Same as Fig. 2.5.1 but for four subdivisions. Each panel (a)–(e) is for a different type of composite pattern (see Table 2.5.2). ∎

contributions qualitatively. Confirmation of the qualitative discussion can be found in the details of the analysis below. Fig. 2.5.5 illustrates an example of a composite pattern formed from four imprinted patterns in a subdivided network with eight total subdivisions. We will analyze the stability of a neuron in the first subdivision. The state of this neuron is initially chosen from the first imprinted pattern. We must determine ifit retains this value after an update of the network. The synapse matrix contains one contribution from the imprinting of each of the imprinted patterns,and the synapses between subdivisions are reduced by the factor g.

The signal term that tries to maintain the stability of the neuron in the first subdivision arises from the imprint of the first pattern. However, only synapses to other neurons whose activity is set according to the first imprinted pattern contribute to the

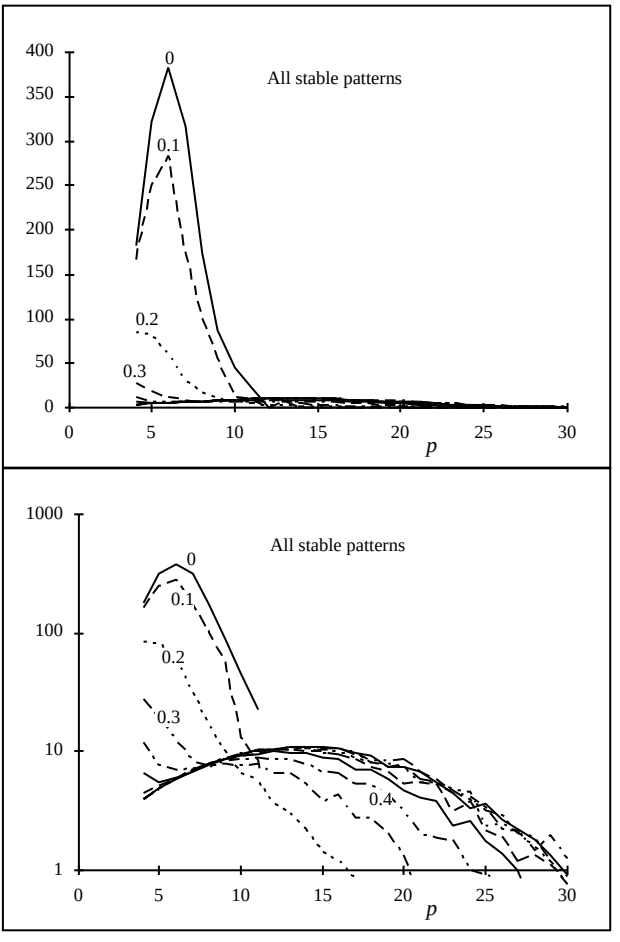

Figure 2.5.4 Total number of stable composite and imprinted patterns corresponding to the sum of Fig. 2.5.2(a)–(e). (a) shows a linear scale, and (b) shows a logarithmic scale for the same results. ∎

signal. Thus, the signal arises from synapses to other neurons within the first subdivision,and also from synapses to neurons in the second subdivision, but not to neurons in any other subdivisions. The signal from the second subdivision is reduced from what it would be in a fully connected network by the factor g.

The noise terms arise from the imprinting of all the other patterns. However, there are special problems with the subdivisions that have other patterns present in them. These subdivisions are trying to recreate their own pattern in the full network. For example,the third, fourth and fifth subdivisions in Fig. 2.5.5 all contain the sec-

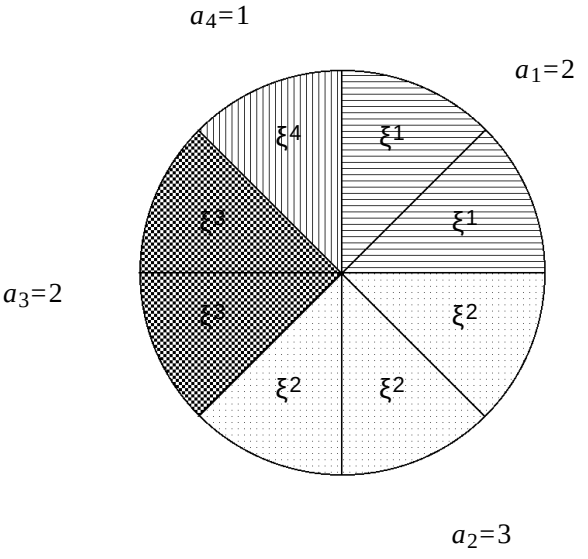

Figure 2.5.5 Schematic illustration of a composite pattern represented by a subdivided network. Each of the distinct shadings indicates the region of the network that contains a particular imprinted pattern. In the illustrated case, the network has eight subdivisions and the composite pattern is formed from parts of four imprinted patterns, ξ^1 through ξ^4. For simplicity the parts of the network that contain the same pattern are shown adjacent to each other. The values of a_1 through a_4 indicate how many subdivisions represent each of the imprinted patterns. ∎

ond imprinted pattern. The neurons in these subdivisions act coherently to try to influence the neuron in the first subdivision to take its value according to the second imprinted pattern. Its value in the first imprinted pattern, whose stability we are testing, may or may not be the same as its value in the second imprinted pattern.On average,half of the neurons in the first subdivision will receive an influence that will try to flip them. Because all the neurons in the third, fourth and fifth subdivisions act coherently to try to reconstruct their pattern in the first subdivision, we must calculate their combined influence as a contribution to the noise which may destabilize the pattern in the first subdivision. It is important that this destabilizing influence is diminished by the factor g, since in a fully connected network the composite pattern would be unstable for this reason.

An important distinction between two cases arises in our analysis when we consider the combined effect of all of the other patterns present in the composite state. There are a total of four patterns in Fig. 2.5.5. This means that there are three coherent noise terms that are trying to destabilize the first pattern. When we calculate the effect of these noise terms, we must decide whether we can average them together or whether we must add their effects. The correct answer depends on how many patterns

there are. When there are only a few patterns, we cannot average them together, but when there are many, we can. By the analysis discussed below, the crossover point occurs roughly at seven patterns.A simple way to understand this result is to realize that seven equal contributions to the noise will add together (have the same sign) 1 in 2^7 times, or just under 1% of the time. As discussed in the case of a fully connected network (Section 2.2.5),this is the limiting fraction of unstable neurons that can be tolerated in a stable pattern. Thus, when there are more than seven patterns,it is not necessary to add their contributions to determine the impact of the pattern stability; it is enough to average them.

The existence of a crossover in the behavior of the subdivided network with seven parts of a composite pattern is the basis of our discussion of the 7 ± 2 rule. In essence, when we can average over the effects of other subdivisions,then each subdivision does not influence the other subdivisions directly, only the average effect is relevant. In contrast, when there are no more than seven different patterns, which is always true when there are no more than seven subdivisions,then the effect of each of the subdivisions must be considered explicitly in evaluating the stability.

We review and introduce additional notation for the signal-to-noise analysis. We assume a network comprised of q subnetworks each containing $N' = N/q$ neurons that are fully internally connected but more weakly connected to each other. The ratio of connection strengths is controlled by the parameter $g \quad [0,1]$. $g = 0$ corresponds to a completely subdivided network and $g = 1$ corresponds to a fully connected network. For arbitrary g, the synaptic connection matrix is written as:

$$J_{ij} = \begin{cases} J'_{ij} & \left\lfloor \frac{i}{N'} \right\rfloor = \left\lfloor \frac{j}{N'} \right\rfloor \\ gJ'_{ij} & \text{otherwise} \end{cases} \tag{2.5.10}$$

where $\lfloor x \rfloor$ is the integer part of x. The first case corresponds to i and j in the same block along the matrix diagonal,i.e.,in the same subnetwork. J' is the usual Hebbian matrix:

$$J'_{ij} = \begin{cases} \frac{1}{N} \sum_{\mu=1}^{p} \xi_i^\mu \xi_j^\mu & i \neq j \\ 0 & i = j \end{cases} \tag{2.5.11}$$

The composite pattern that we wish to test the stability of is formed out of pieces of imprinted states. As in the simulations, it is important to distinguish how many subdivisions have the same imprinted pattern. We test the stability of a trial composite pattern written in the form:

$$\left(\xi_1^1 \quad \cdots \quad \xi_{N'a_1}^1 \quad \xi_{N'a_1+1}^2 \quad \cdots \quad \xi_{N'(a_1+a_2)}^2 \quad \cdots\right) \tag{2.5.12}$$

This pattern is constructed by taking the first a_1 subdivisions from the corresponding part of the first pattern. The next a_2 subdivisions are taken from the second pattern

and so on. In general there are a_μ subdivisions extracted from the pattern ξ^μ. We denote the number of $\{a_\mu\}$ that are non zero by $\hat{p}$. The sum over all a_μ is the total number of subdivisions:

$$\sum_{\mu=1}^{\hat{p}} a_\mu = q \tag{2.5.13}$$

We start by considering the stability of the first subdivision by looking at the stability of the first neuron

$$s_1 h_1 = s_1 \sum_{j=2}^{N} J_{1j} s_j \tag{2.5.14}$$

and separate J into the part due to the subdivision itself and all the rest, which is multiplied by the factor g compared to the usual expression:

$$= \frac{1}{N} \sum_{j=2}^{N'} \sum_{\mu=1}^{p} s_1 \xi_1^\mu \xi_j^\mu s_j + \frac{g}{N} \sum_{j=N'+1}^{N} \sum_{\mu=1}^{p} s_1 \xi_1^\mu \xi_j^\mu s_j \tag{2.5.15}$$

It is helpful to consider separately the part of the j sum that corresponds to each of the parts of the composite pattern. The first imprinted pattern appears in the composite pattern not only in the first subnetwork, but in the first a_1 subnetworks. We also have to consider each of the other patterns in the composite state. To simplify the notation, we will look at only the second pattern (in the subdivisions $\{a_1 + 1,...,a_1 + a_2\}$) and add the rest at the end:

$$= \frac{1}{N} \sum_{j=2}^{N'} \sum_{\mu=1}^{p} s_1 \xi_1^\mu \xi_j^\mu s_j + \frac{g}{N} \sum_{j=N'+1}^{a_1 N'} \sum_{\mu=1}^{p} s_1 \xi_1^\mu \xi_j^\mu s_j + \frac{g}{N} \left[\sum_{j=a_1 N'+1}^{(a_1+a_2)N'} \sum_{\mu=1}^{p} s_1 \xi_1^\mu \xi_j^\mu s_j + \ldots \right] \tag{2.5.16}$$

With this separation of the sum over j, we can now replace the values of s_j with their corresponding values in terms of the imprinted patterns. The first two sums have $s_j = \xi_j^1$ and the third term has $s_j = \xi_j^2$. We also substitute the value of $s_1 = \xi_1^1$ which appears in each of the terms:

$$= \frac{1}{N} \sum_{j=2}^{N'} \sum_{\mu=1}^{p} \xi_1^1 \xi_1^\mu \xi_j^\mu \xi_j^1 + \frac{g}{N} \sum_{j=N'+1}^{a_1 N'} \sum_{\mu=1}^{p} \xi_1^1 \xi_1^\mu \xi_j^\mu \xi_j^1 + \frac{g}{N} \left[\sum_{j=a_1 N'+1}^{(a_1+a_2)N'} \sum_{\mu=1}^{p} \xi_1^1 \xi_1^\mu \xi_j^\mu \xi_j^2 + \ldots \right] \tag{2.5.17}$$

The signal will arise from the imprinting of the first pattern, $\mu = 1$, but only in the first two terms above. All the rest of the terms will give rise to the noise. However, we must also be careful in the third term how we treat the contribution of the imprinting of the second pattern $\mu = 2$. So we separate these parts from each of the terms:

$$= \frac{1}{N}\sum_{j=2}^{N'} \xi_1^1\xi_1^1\xi_j^1\xi_j^1 + \frac{g}{N}\sum_{j=N'+1}^{a_1N'} \xi_1^1\xi_1^1\xi_j^1\xi_j^1$$
$$+ \frac{1}{N}\sum_{j=2}^{N'}\sum_{\mu=2}^{p} \xi_1^1\xi_1^\mu\xi_j^\mu\xi_j^1 + \frac{g}{N}\sum_{j=N'+1}^{a_1N'}\sum_{\mu=2}^{p} \xi_1^1\xi_1^\mu\xi_j^\mu\xi_j^1 \qquad (2.5.18)$$
$$+ \frac{g}{N}\left[\sum_{j=a_1N'+1}^{(a_1+a_2)N'} \xi_1^1\xi_1^1\xi_j^1\xi_j^2 + \sum_{j=a_1N'+1}^{(a_1+a_2)N'} \xi_1^1\xi_1^2\xi_j^2\xi_j^2 + \sum_{j=a_1N'+1}^{(a_1+a_2)N'}\sum_{\mu=3}^{p} \xi_1^1\xi_1^\mu\xi_j^\mu\xi_j^2 + \ldots\right]$$

We can now resolve all squares of variables to +1. When we do this,the first two sums can be directly evaluated, since they are sums over unity. Note also that the middle sum in the square brackets is a summation over terms that are independent of j, and can be evaluated by taking out the factor $\xi_1^1 = \xi_1^2$:

$$= \frac{N'-1}{N} + \frac{g(a_1-1)N'}{N}$$
$$+ \frac{1}{N}\sum_{j=2}^{N'}\sum_{\mu=2}^{p} \xi_1^1\xi_1^\mu\xi_j^\mu\xi_j^1 + \frac{g}{N}\sum_{j=N'+1}^{a_1N'}\sum_{\mu=2}^{p} \xi_1^1\xi_1^\mu\xi_j^\mu\xi_j^1 \qquad (2.5.19)$$
$$+ \frac{g}{N}\left[\sum_{j=a_1N'+1}^{(a_1+a_2)N'} \xi_j^1\xi_j^2 + \xi_1^1\xi_1^2 a_2N' + \sum_{j=a_1N'+1}^{(a_1+a_2)N'}\sum_{\mu=3}^{p} \xi_1^1\xi_1^\mu\xi_j^\mu\xi_j^2 + \ldots\right]$$

The signal term is now visible. The first part of the signal arises from the first subdivision acting upon itself,and the rest is from the other subdivisions that contain the first imprinted pattern. We can take $N >> 1$ and substitute $q = N/N'$ to obtain the expression:

$$\text{signal} = \frac{(1-g)+ga_1}{q} \qquad (2.5.20)$$

To evaluate the noise we must pay attention to the special terms mentioned before. We have succeeded to resolve Eq.(2.5.19) in such a way that each of the remaining terms is uncorrelated in sign.When we reached this point in the uniform network, all we had to do was to count the number of terms—the number of steps in the random walk—and use a root mean square evaluation of its magnitude. In this case, however, all of the steps do not have the same magnitude. This is a particular problem for the special term in the middle of the square bracket. There will be one such term for each of the imprinted patterns. We can rewrite the noise term by replacing the uncorrelated values with the notation ±1. This makes it easier to count how many terms there are in each sum:

$$\frac{1}{N}\sum_{j=2}^{N'}\sum_{\mu=2}^{p}(\pm 1)+\frac{g}{N}\sum_{j=N'+1}^{a_1N'}\sum_{\mu=2}^{p}(\pm 1)+\frac{g}{N}\left[\sum_{j=a_1N'+1}^{(a_1+a_2)N'}(\pm 1)+(\pm 1)a_2N'+\sum_{j=a_1N'+1}^{(a_1+a_2)N''}\sum_{\mu=3}^{p}(\pm 1)+\ldots\right]$$

$$=\frac{1}{N}\sum^{(N'-1)(p-1)}(\pm 1)+\frac{g}{N}\sum^{(a_1-1)N'(p-1)}(\pm 1)+\frac{g}{N}\left[\sum^{a_2N'}(\pm 1)+(\pm 1)a_2N'+\sum^{a_2N'(p-2)}(\pm 1)+\ldots\right]$$

$$=\frac{1}{N}\sum^{(N'-1)(p-1)}(\pm 1)+\frac{g}{N}\sum^{(a_1-1)N'(p-1)}(\pm 1)+\frac{g}{N}\sum_{\nu=2}^{\hat{p}}\left[\sum^{a_\nu N'}(\pm 1)+(\pm 1)a_\nu N'+\sum^{a_\nu N'(p-2)}(\pm 1)\right] \tag{2.5.21}$$

In the last expression we have restored the contribution of all of the other parts of the composite pattern explicitly. We can now see that there are three kinds of steps in our random walk,those that have a coefficient of $1/N$ of which there are $(N'-1)(p-1)$, those with a coefficient of g/N of which there are a total of

$$\begin{aligned}(a_1-1)N'(p-1)+\sum_{\nu=2}^{\hat{p}}\left(a_\nu N'+a_\nu N'(p-2)\right)&=qN'(p-1)-N'(p-1)\\&=(N-N')(p-1)\end{aligned} \tag{2.5.22}$$

and a special set of $\hat{p}-1$ terms with coefficients of the form

$$\frac{ga_\nu N'}{N}=\frac{ga_\nu}{q} \tag{2.5.23}$$

Because of this last set of terms—the coherent noise terms—we have to be much more careful about our analysis than in the uniform network case. The magnitude of the first two kinds of terms are small as N becomes large, and the number of terms is large. The coherent noise terms may be large even for large N, and their number can be small since it only increases with $\hat{p}$. For the uniform network we considered a root mean square estimate of the noise. This root mean square estimate only works,however, if the number of terms is large. Thus we must distinguish between the cases where $\hat{p}$ is small and when $\hat{p}$ is large.

When we studied the signal-to-noise analysis of the fully connected network and the retrieval of imprinted patterns, we found that for low storage, $p << N$, the noise disappeared and retrieval of the imprinted patterns would occur. Then we could consider the storage capacity as p increased. For composite patterns, the situation is different, because the noise does not disappear for low storage. Thus we first study the low-storage case.

We start by considering two estimates of the magnitude of the noise relevant for $\hat{p}$ large and small respectively. For the case of $\hat{p}$ large we can use a root mean square estimate of the noise because the number of independent terms is large. To obtain the noise we use a generalization of the random walk with different step sizes D_i (left half of Eq. 1.2.51):

$$\sigma=\sqrt{\sum D_i^{\,2}} \tag{2.5.24}$$

In the limit of low storage $p << N$ and not too many subdivisions $q << N$ only the coherent noise terms are important. From Eq.(2.5.23) this gives us a root mean square noise

$$\text{noise} \approx \left(\frac{g}{q}\right)\sqrt{\sum_{\nu \neq 1} a_\nu^2} \tag{2.5.25}$$

When $\hat{p}$ is small we cannot average the noise terms and consider only the typical value. Thus we consider the maximum possible effect of the coherent noise terms. This occurs when all values of ± 1 are -1. The magnitude of the maximum noise is

$$\text{maximum noise} = \frac{g}{q}\sum_{\nu=2} a_\nu = \frac{g}{q}(q - a_1) \tag{2.5.26}$$

If the signal is greater than the maximum noise, then the pattern must be stable. However, if the signal is significantly greater than the typical noise, but less than the maximum noise, then it may be stable. If the signal is about the same as the typical noise, then the pattern is almost certainly unstable.

Thus, we can guarantee retrieval of a composite pattern if the maximum possible noise is less than the signal. This places a limit on g determined by the inequality

$$\frac{(1-g)+ga_1}{q} > \frac{g}{q}(q - a_1) \tag{2.5.27}$$

If $g = 0$, this inequality is always satisfied. This is just the completely subdivided case that we know leads to stability of composite patterns. If $g = 1$,this condition becomes:

$$a_1 > q/2 \tag{2.5.28}$$

This means that the first pattern must have more than half of the subdivisions in order to be stable. How we define subnetworks is arbitrary in the case of $g = 1$, but the meaning of this statement is that a pattern is stable only if it occupies more than half of the network. However, this must apply to each of the parts of the composite pattern,and thus implies that no composite pattern except the trivial one of a single imprinted pattern can be stable in the case $g = 1$.

If we assume that $1 > g > 0$, we can simplify the inequality in Eq. (2.5.27) to obtain:

$$\frac{(1-g)}{g} > (q - 2a_1) \tag{2.5.29}$$

or

$$g < \frac{1}{q - 2a_1 + 1} \tag{2.5.30}$$

This limit on g ensures that the part of the composite pattern in the first subnetwork is stable. In order for all parts of the composite pattern to be stable, g must be smaller than the minimum value of this expression taken over all subnetworks:

$$g < \min_{\mu} \frac{1}{q - 2a_{\mu} + 1} = \frac{1}{q - 2a + 1} \tag{2.5.31}$$

where

$$a = \min a_{\mu} \tag{2.5.32}$$

is the minimum number of subnetworks containing any one of the imprinted patterns. The greatest restriction on g arises from demanding the stability of the smallest part of the composite pattern (smallest a_{μ}). This result is the same as Eq. (2.5.3).

For $\hat{p}$ large, this limit on g is overly severe, since the maximum noise occurs infrequently. In this case we use the root mean square estimate of the noise, Eq.(2.5.26). However, stability of the pattern does not result when the signal is just greater than the noise. As in the signal-to-noise analysis of the fully connected network (Section 2.2.5),it must be sufficiently greater to ensure that only about 1% of the neurons will be unstable. The mean probability that a given neuron is unstable is given by integrating the probability of the noise being less than the signal. We thus require the signal-to-noise ratio to be greater than the number $r = 1/\sqrt{\alpha_c} \approx 2.64$:

$$\frac{ga_1 + (1-g)}{q} > r\frac{g}{q}\sqrt{\sum_{\lambda \neq 1} a_{\lambda}^{\ 2}} \tag{2.5.33}$$

This gives a limit on acceptable g of

$$g < \frac{1}{r\sqrt{\sum_{\lambda \neq 1} a_{\lambda}^{\ 2}} - a_1 + 1} \tag{2.5.34}$$

If we assume that the maximal allowable error rate at any neuron in any subdivision is given by this inequality, then we have the result:

$$g < \frac{1}{r\sqrt{\sum_{\lambda} a_{\lambda}^{\ 2} - a^2} - a + 1} \tag{2.5.35}$$

The limit in Eq.(2.5.35) corresponds to a certain probability, rather than a guarantee, that subnetworks of the composite pattern are stable. This is important,since requiring that all parts of the composite pattern are likely to be stable is a much stricter condition. Similarly, requiring that at least one of the parts is likely to be stable is a much weaker condition. For example,if all of the subdivisions have distinct imprinted patterns $\{a_{\mu} = 1\}$ and each has a probability P of being stable then the probability that all are stable is only P^q and the probability that at least one is stable is $1 - (1 - P)^q$. Thus, as a function of g, composite patterns become progressively unstable in more subdivisions in the vicinity of the limit in Eq. (2.5.35).

The analysis that we have performed reveals an interesting limitation to the degree of useful subdivision if we consider the role subdivision may play in the brain, or

in an artificial pattern-recognition task. The second limit we obtained in Eq.(2.5.35) is a higher one than the first limit in Eq. (2.4.37) when there are many subdivisions. Why is this a problem? Because it indicates that once the number of subdivisions becomes large, for g values that satisfy Eq.(2.5.35),essentially all possible combinations become stable, but some parts will be stable at higher values of g and some at lower values. Moreover, individual subdivisions do not affect the stability of the state. In contrast, when there are only a few subdivisions, we can control which combinations are stable using the value of g, and the state of each subdivision matters.

We can estimate the crossover point where the number of subdivisions becomes too large by looking at a pattern where each subdivision has a different imprinted pattern $a_\mu = 1$, for $\mu = 1,...,q$, and equating the two limits

$$\frac{1}{r\sqrt{q-1}-1+1} = \frac{1}{q-2+1} \tag{2.5.36}$$

or

$$q = r^2 + 1 \approx 7.94 \tag{2.5.37}$$

This suggests that q should be less than this value for effective use of partially correlated information stored by subdivisions in a network. It is possible to suggest that this limitation in the number of useful subdivisions may be related to the characteristic number of independent pieces of information a human is able to "keep in mind"at one time. This number is conventionally found to be 7 ± 2. The comparative rigidity of this number as compared with many other tests of variation between human beings suggests that it may indeed be tied to a fundamental limitation related to the architecture of neural networks as we have found.

Up to this point we have been considering the case of low storage. In the case of high storage, we must consider all three types of noise terms found in Eq. (2.5.21) rather than just the coherent terms. We should still distinguish between the cases where $\hat{p}$ is small or large. In either case, we estimate the contribution of the first two types of terms as a random walk. However, only for $\hat{p}$ large can we treat the coherent terms as a random walk.

For $\hat{p}$ large we calculate the typical noise from the three types of terms as:

$$\text{noise} \approx \sqrt{\left(\frac{1}{N}\right)^2 (N'-1)(p-1) + \left(\frac{g}{N}\right)^2 (N-N')(p-1) + \left(\frac{g}{q}\right)^2 \sum_{\nu\neq 1} a_\nu{}^2} \tag{2.5.38}$$

This can be simplified using $N, N', p >> 1$ to obtain:

$$\text{noise} \approx \sqrt{\frac{(1+g^2(q-1))}{q}\frac{p}{N} + \left(\frac{g}{q}\right)^2 \sum_{\nu\neq 1} a_\nu{}^2} \tag{2.5.39}$$

The relationship between the storage capacity, which limits the value of p/N, and the interconnection strength g can be found by setting the signal-to-noise ratio to be less than $r = 1/\sqrt{\alpha_c}$. This gives

$$p < \frac{N\left(\alpha_c(1-g+ga)^2 - g^2\sum a_\nu{}^2 - g^2a^2\right)}{q(1+g^2(q-1))} \tag{2.5.40}$$

We have replaced a_1 by a. Note that the numerator is zero when the maximum value of g, according to Eq. (2.5.35), is reached.

For $\hat{p}$ small we must take the maximum value of the coherent noise terms. This value should be subtracted from the signal before we compare the result with the root mean square value of the rest of the noise. Separating the two noise terms from each other we have:

$$\text{noise}_1 = \frac{g}{q}(q-a_1) \tag{2.5.41}$$

$$\text{noise}_2 = \sqrt{\left(\frac{1}{N}\right)^2 (N'-1)(p-1) + \left(\frac{g}{N}\right)^2 (N-N')(p-1)} \tag{2.5.42}$$

or using $N, N', p >> 1$

$$\text{noise}_2 = \sqrt{\frac{p}{qN}}\sqrt{1+g^2(q-1)} \tag{2.5.43}$$

Subtracting the first noise term from the signal and insisting the result is greater than r times the rest of the noise we have:

$$\frac{(1-g)+ga_1}{q} - \frac{g}{q}(q-a_1) > r\sqrt{\frac{p}{qN}}\sqrt{1+g^2(q-1)} \tag{2.5.44}$$

This implies that the number of patterns that can be imprinted,and the corresponding composite patterns recalled, is limited by:

$$p < \frac{\alpha_c N}{q}\frac{\left(1-g-gq+2ga\right)^2}{1+g^2(q-1)} \tag{2.5.45}$$

where we have again replaced a_1 by a. Note that the numerator is zero when the maximum value of g at low storage is reached according to Eq. (2.5.32). The storage capacity increases for lower values of g if a is small compared to q; i.e., for a composite pattern. If we ask about the retrieval of only the imprinted patterns, then we can use the same expression with $a = q$ or

$$p < \frac{\alpha_c N}{q}\frac{(1-g+gq)^2}{1+g^2(q-1)} \tag{2.5.46}$$

In this case the maximum storage is for $g = 1$, as we would expect for the imprinted patterns.

Question 2.5.1 Compare the signal-to-noise analysis presented above with a signal-to-noise analysis of a subdivided network where the subdivision is accomplished by diluting inter-subnetwork synapses.Specifically, assume that a fraction g' of synapses between neurons in different subdivisions remain after dilution.

Solution 2.5.1 Since the signal-to-noise analysis of a diluted network follows very similar steps to the analysis of a network whose inter-subnetwork synapses are multiplied by the factor g, we mention only the differences in the two treatments.

Reducing the number of terms in a sum by the factor g' leads to the same effect on its average value as multiplying each of the terms by the same factor. However, there will be a different effect on a root mean square estimate of the magnitude of a random walk.A random walk with fewer steps, by a factor g', will be reduced in magnitude by a factor of $\sqrt{g'}$ rather than g'.

Thus, the analysis of the signal is the same for dilution as that given in the text except for the substitution of g by g'. This also applies to the low-storage analysis of the coherent noise terms, either for small $\hat{p}$ or for large $\hat{p}$. In each of these cases, the sums over individual synapses are performed directly, rather than as a random walk, and thus g can be directly replaced by g'.

The only place in the analysis where the dilution gives a different result is in the discussion of the noise terms that limit the storage capacity. These terms, in Eq.(2.5.39) and Eq.(2.5.43), can be found for the case of dilution by substituting $\sqrt{g'}$ for g. The resulting noise terms are larger for the case of dilution, resulting in a smaller storage capacity as compared to the effect of multiplying all the inter-subnetwork synapses by the same factor. ∎

2.6 From Subdivision to Hierarchy

In the last section our analysis of the properties of partially subdivided networks led to a conclusion that begs for further discussion. Our motivation for investigating the properties of subdivided networks was to discover the underlying purpose of functional subdivision. We were able to demonstrate that subdivision does provide a mechanism for storage of patterns with a particular composite structure. However, we encountered a fundamental limitation. Once there are too many subdivisions, the ability to store correlations between the subdivisions is diminished. In this section we review the argument that led to this conclusion and discuss further implications.

A fully connected network stores complete neural patterns.On the other hand, a completely subdivided network stores independent subpatterns without correlations between them. For most applications it is reasonable to assume that different aspects of the information are partially independent. This requires the ability to balance the storage of independent subpatterns and yet retain correlations between them. To achieve this balance requires an intermediate degree of interconnection between the

subdivisions. This is possible, we found, when the number of subdivisions is small. However, when the number of subdivisions is large there is essentially no intermediate possibility: either the connections are strong enough to store complete states or they are weak enough to allow all combinations. What is particularly surprising is that the meaning of the term "large" is any number greater than roughly seven. While this is consistent with the well established 7±2 rule of short term memory, the implications extend yet further.

We are limited to a maximum of seven subdivisions, and yet the advantages of subdivisions for storage of independent aspects of information extends to many more subdivisions. An architecture that could provide further use of subdivision within the limitation is a hierarchically subdivided system. By keeping the branching ratio less than seven, we would construct a network formed of small networks that are strongly coupled, large networks that are weakly coupled, and still larger networks that are more weakly coupled. At each level of organization the strength of the connections within each subdivision must be strong enough compared to the connections between subdivisions to establish the integrity of the subdivision. Yet they must be weak enough to allow for the influence of the other subdivisions.Our model of the brain is no longer a model of interacting neurons, but rather of interacting units that at each level of organization attain an additional degree of complexity.

The brain has been found to be subdivided in a hierarchical fashion, and at least at the level of the major structures,this hierarchy does not have a high branching ratio. The brain is formed from the cerebrum, the cerebellum and the brain stem. The cerebrum is divided into two hemispheres. Each hemisphere is divided into four lobes.Each lobe is further divided into smaller functional regions;however, there are fewer than ten of these per lobe. The brain stem can also be subdivided into a hierarchy of functional regions. One could argue that the mapping of these structures reflects our own abilities caused by the 7±2 rule that lead us to divide the brain into only a few parts when we study it. This, however, misses the point of our observations. Our conclusions predict the relative strength of interdependence of different sections of the brain. This prediction would require additional systematic studies to confirm.

As we emphasized in the introduction to this book, our approach is primarily a statistical one. However, if the system we are investigating is composed out of only a few distinct components,then the statistical approach must have limited ability to describe it. When there are only a few components,each one should be specifically designed for the purpose to which it is assigned. A general statistical approach does not capture their specific nature. This is the reason the preface recommends the need for complementary investigation of particular aspects of individual complex systems. Here, we will continue to pursue other general principles through the statistical study of these systems. It is natural to ask whether we can generalize the conclusions from the study of neural networks to other complex systems. Several aspects of this question are discussed in the following section and others will arise in later chapters of this book.

2.7 Subdivision as a General Phenomenon

In this section we pursue the question of the necessity of subdivision and substructure in complex systems. All of the examples of complex systems that we will discuss in this text have the property that they are formed from substructures that extend to essentially the same scale as the whole system. We will review them as they are introduced more fully later in the text. Here we summarize some examples. The human body has nine physiological systems further divided into organs. Proteins have substructure formed of α-helices and β-sheets, and are often organized together with other proteins in quaternary structures.Life on earth considered as a complex system is divided among climates, ecosystems,habitats and species. Weather is formed out of large-scale air and ocean currents,storms and regions of high and low pressure. In all of these systems the largest scale of subdivision comprises fewer than 100 parts,and more typically of order 10 parts of the whole system. Why should this be the case?

Our discussion of subdivision in this chapter has been based on the function of the neural network as a memory. The importance of both combinatorial expansion of composite states and the constraints upon them due to interactions between subnetworks played a role. Similar considerations may apply in some of the other complex systems. For example,in the immune system the importance of composite states is apparent in the generation of a large variety of immune receptors by a process that combines different segments of DNA.A related discussion of substructure in the context of evolution will be included in Chapter 6.

In this section we adopt a different approach and relate substructure to the categories of complex systems that were articulated in Section 1.3. We argue qualitatively that substructure is necessary for what we generally consider to be a complex system. More specifically, we distinguish between a complex material and a complex organism. As defined in Section 1.3, a complex material is a system from which we can cut a part without significantly affecting the rest of the system.A complex organism is a system whose behavior depends on all of its parts and so is changed when a piece is removed. We propose that complex organisms require substructure.

In a system formed out of many interacting elements,each element may interact directly with a few or with many other elements.Our concern is to establish the conditions that imply that the behavior of a particular element depends on various sets of elements that comprise a significant fraction of the system. When this is true we have a complex organism. Otherwise,parts of the system may be removed without affecting the rest,and the system is a complex material or the system may even be a divisible thermodynamic system. The effective interaction between elements may be direct, or may be indirect because it is mediated by other elements. However, even if there is a direct interaction,ifit does not affect the behavior of the element we are considering, then this interaction is irrelevant as far as we are concerned.

Let us start by considering generic interacting spin models such as the Ising model (Section 1.6). When the interaction between spins is local, then the system is generally a divisible thermodynamic system. When the interactions are long range,

then,if there is a dominant ground state, we have a divisible thermodynamic system analogous to a magnet. If there are competing ground states,the system behaves as a spin glass or as an attractor neural network model with trained states. The neural network is the most favorable for consideration as a complex system.

We classify a fully connected neural network with Hebbian imprinting as a complex material rather than as a complex organism. This classification is based upon evaluating the impact of removing, say, 10% of the neurons.Our main concern in describing a neural network is the storage and retrieval of patterns. If we have only a few imprinted states, then separating the smaller part of the network does not affect the ability of either the large or small parts to retrieve the patterns. This is characteristic of a divisible system. If the number of stored patterns p is greater than the capacity of the smaller part, by itself, but smaller than the capacity of the larger part ($0.9\alpha_c N > p > 0.1\alpha_c N$), then the smaller part will fail in retrieval and the larger part will be unaffected. This is the regime of operation in which the network would be expected to be used—the regime in which its storage capacity is utilized and the basins of attraction remain significant. The behavior is characteristic of a complex material. On the other hand,if we are very close to the full capacity of the network ($\alpha_c N > p > 0.9\alpha_c N$) then both the large and small parts of the network will be affected by the removal of the small part. We could consider this to be a regime in which the network has the behavior of a complex organism. However, this is the regime in which the basins of attraction of memories are significantly degraded,and any perturbation affects the performance of the system. A better way to approach the classification problem is to consider the number of states that can be stored before and after the separation. We see that the storage capacity of the larger part of the system is weakly affected by the removal of a small part, while the small part is strongly affected. Thus the fully connected network should be classified as a complex material.

We classified the attractor network as a complex material on the basis of our investigations of its properties. However, the reason that the system is not a complex organism rests more generally on the existence of long- (infinite-) range interactions. If a particular element of the system interacts with all other elements, it is difficult for the removal of 10% of the system to affect its behavior significantly. Since there is nothing that differentiates the part of the system that was removed from any other part,the most that can be affected is 10% of the forces that act on the particular element. This is not enough, under most circumstances, to affect its behavior.

We found that short-range or long-range interactions do not give rise to complex organism behavior. Since an element cannot be affected by many other elements directly, it makes sense for us to start with a model where the element is primarily affected by only a few other elements that constitute its neighbors. This is the best that can be achieved. Then,so that it will be affected by other elements, we arrange interactions so that the neighborhood as a whole is affected by several other neighborhoods, and so on in a hierarchical fashion. This is the motivation for substructure.

What happens if we cut out one subdivision of a subdivided network which has only a few subdivisions, or a significant fraction of a subdivision? The stable states of the network are composite states. If the interactions between subdivisions are too

weak,then all composite states are stable and removal will affect nothing. The system is a completely divisible system. If the interactions are too strong, then we are back to the case of a fully connected network. However, in the intermediate regime determined in Section 5.4.5, the stability of the composite states depends on the interactions between the subdivisions and the behavior of the large and small parts are both affected. Why? The reason is that only some of the composite patterns are stable. Which ones are stable depends on all of the subdivisions and their interactions. Thus, in the intermediate regime of interactions,the system behaves as a complex organism.

We could further develop the complex organism behavior of a subdivided network by recalling that the architecture is designed so that a particular aspect of the information is present in each subdivision. The loss of a subdivision would cause the loss of this aspect of the information. While this is reasonable argument, it is not a fundamental difference between a subdivided system and the fully connected network. We could choose to map different aspects of the information onto different parts of a fully connected network and arrive at the same conclusion. Even though this is more natural for a subdivided system,it is not inherent in the subdivision itself and therefore does not advance our general discussion.

An alternate way to consider the complex behavior of the subdivided network is in terms of the growth in the number of stable states of the network. For a fully connected network, the growth is linear. For a completely subdivided network, the growth is exponential, reflecting the independence of subdivisions. In the intermediate regime of interconnection,the growth in the number of composite states requires more detailed study and depends on the particular way in which the growth is performed. This suggests a level of control of properties of the system by its structure that we associate with a complex organism.

We have been considering the influence of elements of an Ising model upon each other. There is an important case that we have not included that could be represented by a feedforward network or by an Ising model sensitive to boundary conditions. In such systems,the influence of one neuron is transferred by a sequence of steps to other neurons down a chain of influence. We could consider an extreme case in which there is a long sequence of elements each affecting the subsequent one in the chain. Removing any of the elements of this sequence would break the chain of influence and all of the downstream elements would be affected. As a complex system that cannot be separated,this violates our claim that substructure is necessary or necessarily limited to only a few elements.

This counterexample has some validity; however, the argument is not as simple as it might appear. There are two general cases. Either each of the elements in the chain of influence serves only as a conduit for the information, in which case the nature of its influence is minimal, or, alternatively, each element modifies the information being transmitted, in which case generally the influence of the input dies rapidly with distance and only a few previous elements in the sequence are important for any particular element. The former case is like a pipe. All the segments of the pipe are essential for the transmission of the fluid, but they do not affect its nature. From the point

of view of describing the behavior of complex systems,a conduit performs only a single function and therefore may be described using a single (or even no) variables.On the other hand, when each element affects the information, then we are back to the circumstance we have considered previously, and substructure is necessary. The reason that the influence dies with distance along the chain can be understood by considering a sequence of filters. Unless the filters are matched,then even a few filters will block all transmission.

We can revive the counterexample by considering a system whose elements represent a long sequence of logical instructions such as might be found in a computer program. Here we are faced with a problem of interpretation. If the system always represents only one particular program, then it is similar to a conduit. If the program changes, then we must consider the mechanism by which it is changed as part of the system. Nevertheless,the recognition that a narrowly construed sequence of instructions does represent an exception to the 7±2 rule about substructure can play a role in our understanding of the behavior of complex systems.

The discussion of substructure may be further generalized by considering elementary building blocks that are more complex than binary variables. Our objective is to argue that even when the building blocks are highly complex,the generalized 7±2 rule that requires substructure applies without essential modification. It applies therefore to complex systems formed from abstract or realistic neurons or to complex social systems of human beings. Our discussion has already, in a limited sense, considered elements of varied complexity, because subnetworks may contain different numbers of neurons. Our conclusion about the number of allowed subdivisions (seven) was independent of the number of neurons in the subdivision. Why should this be true?

We might imagine that a particular element with a greater complexity can be affected in one way by some elements and in another way by other elements. This would enable the whole complex system to be composed of a number of subdivisions equal to the number of aspects of the element that can be affected.Or we could even allow seven subdivisions for each aspect of the element. Then the number of subdivisions could be the product of seven times the number of aspects of an element. For example, when we think of human physiology, a cell requires both oxygen and sugars for energy production. Why couldn't we construct a system that is composed of seven subdivisions that are involved in providing oxygen and seven subdivisions that are involved in providing sugar, with a result that we have a system of fourteen subdivisions that is still a complex system. We could argue that the oxygen system and the sugar system are new subsystems that are described by the model. However, since there appears to be no limit to the number of aspects of an element,there should be no characteristic limit to the number of relevant subsystems. Make a list of all of the different chemicals required by a cell and generate a separate system for each one.

This argument,however, does not withstand detailed scrutiny. The central point is illustrated by considering a system formed out of elements each of which consists of two binary variables called TOP and BOTTOM. Each of the binary variables is part

of a neural network that is composed out of seven subdivisions. How many subdivisions would there be altogether? The simplest case would be if there were fourteen subdivisions seven of which contain all of the TOP variables and seven of which contain all of the BOTTOM variables. Many other possibilities could be imagined. From the point of view of a complex system, however, as long as the two binary variables at each element behave independently, we could separately consider one set of seven subdivisions by themselves and the other seven subdivisions by themselves. They are completely decoupled. The physical proximity of the two binary variables as part of the same element does not affect the essential independence of the decoupled systems. As soon as there is a coupling between them we are back to where we started from, with interacting binary variables. Thus, increasing the complexity of the elements from which the complex system is composed does not appear to be able to change qualitatively the requirements of substructure found for the neural network model.