CHAPTER

7

AGGREGATING SIGNALS

There are enormous differences between biological systems and the kind of systems we find when we start digging inside a computer or looking at the schematic for a digital microprocessor chip. The most striking contrast between the two paradigms is the way in which signals are combined to form other signals. In a digital system, signals are combined by **logic elements** or **gates**. These elements take several inputs and form the logical AND, NOR, or similar functions. Each logic function can be thought of as making a *decision* about the inputs, and as reporting that decision as a binary signal at its output. We can construct any logic function by appropriately connecting two-input inverting gates of a single type— NOR, for example. We can think of a complex function as a decision about many inputs—we can always reach that decision by making partial decisions about a few inputs separately, and then combining these partial decisions into a more global decision. We can always subdivide the decision process in this way without increasing the complexity of the resulting circuit by more than a constant factor.

The nervous system combines inputs according to completely different principles. A very large number of inputs are brought together, or **aggregated,** in an analog manner. We can think of the simplest neuron as forming the analog sum of the inputs from the axons and dendrites of other neurons in a tree of passive dendritic processes, such as that shown in Figure 7.1 (see also Figure 4.1 (p. 44)). If enough current is injected into

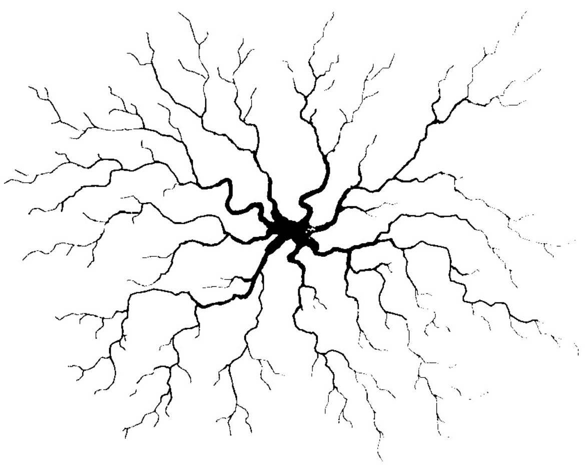


FIGURE 7.1 The dendritic tree of a neuron. This structure may or may not be used to generate action-potential outputs on an axon. Synapses on the tips of the tree provide outputs from, as well as inputs to, the dendritic processes.

the dendritic tree to bring the potential at the axon hillock to a *threshold* value, the nerve will initiate pulses (action potentials) in its axon. We can think of each branch of the dendritic tree as computing the sum of the currents in its two subtrees. Only when these partial sums have been fully aggregated into a total sum can the neuron make a decision to fire a pulse down the axon. In the aggregation process, analog information is preserved carefully. The nerve postpones the decision process until it has performed as much local aggregation as possible.

In addition to the analog aggregation of inputs, the nervous system differs markedly from digital systems in the sheer number of inputs that are factored into a given computation; this number is the **fan-in** of the neuron. The average circuit in a computer has a fan-in of two, three, or four. The fan-in of an average neuron is 1000 to 3000, or even 10,000. There are cells in the cerebellum with several hundred thousand inputs! The corresponding measure representing the number of distinct places to which the output of a computation goes is the **fan-out**. Considered over a system, the fan-in and fan-out must be the same—every input must be some other computation's output, and vice versa. Biologists use the term **convergence** for fan-in, and **divergence** for fan-out. We will use the biology and electrical engineering terms interchangeably.

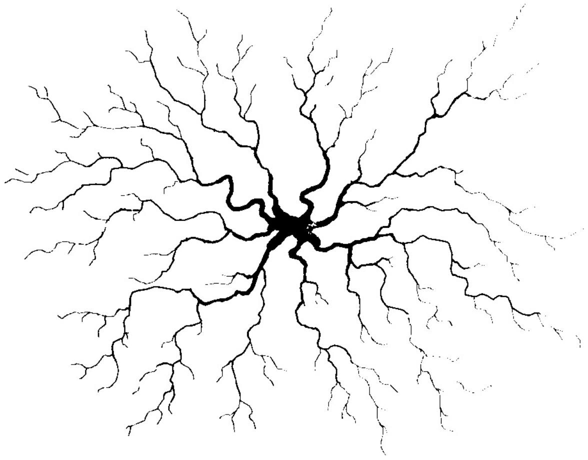
Why do neurons have so many inputs? Each input synapse requires that an axonal process from another neuron be routed to the synaptic site. The more synapses there are, the more wiring is required. In the brain as in silicon, wire fills nearly the entire space. Economizing on wire is the single most important priority for both nerves and chips. At first sight, it would seem that the brain's profligate use of circuits with large fan-in and fan-out is horribly inefficient. As is usual in biological systems, however, there are a number of reasons that compl
to make the high–fan-in neuron a computing device of extraordinary power.

As an example, let us consider a population of neurons that recognize th
presence of specific, reasonably complex objects in the visual field. Neuron A wi
fire if object A is present, neuron B will fire if object B is present, and so on. Th
excitatory synaptic inputs to any particular neuron represent the *features tha*
identify the corresponding object. These features have been computed by lowe
levels of the visual system. A crucial decision among nearly alike objects ofte
is made on the basis of a single feature. The particular feature responsible fo
the decision is dependent on the circumstances: the viewing angle, what part o
the object is obscured by intervening foliage, and so on. Each neuron therefo
must have synaptic inputs representing *all features that might ever be used*, eve
though only a subset of them will contribute to any particular decision. No on
has measured precisely the number of features needed to characterize a comple
object; we will assume that several hundred is a reasonable estimate.

The nervous system represents a single feature not by the output of a singl
neuron, but rather by that of a population of neurons. The presence of a give
feature is represented by the firing of, say, 100 neurons. Not all those neurons fir
at the same rate—those best matched to the particular feature fire at a high rate
whereas those less well matched fire at a lower rate for this feature, but will fir
at a higher rate for a similar but slightly different feature. When a target neuro
requires information concerning the presence of a specific feature, its simples
strategy is to sum all or most of the 100 outputs, *weighting* each input by its firing
rate for the desired feature. For several hundred features, this strategy woul
seem to require several tens of thousands of inputs to a target neuron. Becaus
of the partial overlap of populations representing different features, however, th
number of input synapses can be reduced to perhaps a few thousand.

We can think of the features characterizing a particular object as separat
dimensions of the representation of the object. Any particular recognition tas
will have an **essential dimensionality** corresponding to the number of feature
that, under some circumstances, can become necessary to distinguish two object
A unique recognition cannot be determined if the number of input synapses i
less than the essential dimensionality of the task.

It is intuitive that the precision with which we treat inputs to a computatio
should be related to our confidence in those inputs. It does not pay to comput
the cost of a proposed project to the penny when our estimates of the costs o
several subtasks may be in error by many hundreds of dollars. The details o
any two similar images seen by the visual system are never exactly alike. Even a
familiar object is never seen twice in exactly the same way. We thus have littl
confidence in the details of any particular input. The major task of the nervou
system is to make collective sense of sensory input. Under the conditions impose
by real input data, we can improve the reliability of a decision only by factoring a
large number of inputs into the computation. In other words, when faced with a
decision based on inputs in which it can place little confidence, the brain uses it
resources to *increase the dimensionality* of the computational space, rather tha



E 7.1 The dendritic tree of a neuron. This structure may or may not be used to ate action-potential outputs on an axon. Synapses on the tips of the tree provide s from, as well as inputs to, the dendritic processes.

itic tree to bring the potential at the axon hillock to a *threshold* value, will initiate pulses (action potentials) in its axon. We can think of ch of the dendritic tree as computing the sum of the currents in its ees. Only when these partial sums have been fully aggregated into a can the neuron make a decision to fire a pulse down the axon. In gation process, analog information is preserved carefully. The nerve s the decision process until it has performed as much local aggregation le.

dition to the analog aggregation of inputs, the nervous system differs from digital systems in the sheer number of inputs that are factored en computation; this number is the **fan-in** of the neuron. The average a computer has a fan-in of two, three, or four. The fan-in of an average 1000 to 3000, or even 10,000. There are cells in the cerebellum with undred thousand inputs! The corresponding measure representing the f distinct places to which the output of a computation goes is the **fan-** sidered over a system, the fan-in and fan-out must be the same—every st be some other computation's output, and vice versa. Biologists use **convergence** for fan-in, and **divergence** for fan-out. We will use the nd electrical engineering terms interchangeably.

do neurons have so many inputs? Each input synapse requires that an rocess from another neuron be routed to the synaptic site. The more there are, the more wiring is required. In the brain as in silicon, wire y the entire space. Economizing on wire is the single most important or both nerves and chips. At first sight, it would seem that the brain's e use of circuits with large fan-in and fan-out is horribly inefficient. As is usual in biological systems, however, there are a number of reasons that conspire to make the high–fan-in neuron a computing device of extraordinary power.

As an example, let us consider a population of neurons that recognize the presence of specific, reasonably complex objects in the visual field. Neuron A will fire if object A is present, neuron B will fire if object B is present, and so on. The excitatory synaptic inputs to any particular neuron represent the *features* that identify the corresponding object. These features have been computed by lower levels of the visual system. A crucial decision among nearly alike objects often is made on the basis of a single feature. The particular feature responsible for the decision is dependent on the circumstances: the viewing angle, what part of the object is obscured by intervening foliage, and so on. Each neuron therefore must have synaptic inputs representing *all features that might ever be used*, even though only a subset of them will contribute to any particular decision. No one has measured precisely the number of features needed to characterize a complex object; we will assume that several hundred is a reasonable estimate.

The nervous system represents a single feature not by the output of a single neuron, but rather by that of a population of neurons. The presence of a given feature is represented by the firing of, say, 100 neurons. Not all these neurons fire at the same rate—those best matched to the particular feature fire at a high rate, whereas those less well matched fire at a lower rate for this feature, but will fire at a higher rate for a similar but slightly different feature. When a target neuron requires information concerning the presence of a specific feature, its simplest strategy is to sum all or most of the 100 outputs, *weighting* each input by its firing rate for the desired feature. For several hundred features, this strategy would seem to require several tens of thousands of inputs to a target neuron. Because of the partial overlap of populations representing different features, however, the number of input synapses can be reduced to perhaps a few thousand.

We can think of the features characterizing a particular object as separate *dimensions* of the representation of the object. Any particular recognition task will have an **essential dimensionality** corresponding to the number of features that, under some circumstances, can become necessary to distinguish two objects. A unique recognition cannot be determined if the number of input synapses is less than the essential dimensionality of the task.

It is intuitive that the precision with which we treat inputs to a computation should be related to our confidence in those inputs. It does not pay to compute the cost of a proposed project to the penny when our estimates of the costs of several subtasks may be in error by many hundreds of dollars. The details of any two similar images seen by the visual system are never exactly alike. Even a familiar object is never seen twice in exactly the same way. We thus have little confidence in the details of any particular input. The major task of the nervous system is to make collective sense of sensory input. Under the conditions imposed by real input data, we can improve the reliability of a decision only by factoring a large number of inputs into the computation. In other words, when faced with a decision based on inputs in which it can place little confidence, the brain uses its resources to *increase the dimensionality* of the computational space, rather than to increase the precision with which each individual input is treated. Appendix D provides a formal treatment of several of the points we have mentioned here.

In the balance of this chapter, we will examine examples of networks that use large connectivity to compute interesting functions analogous to several of those computed in the brain.

STATISTICAL COMPUTATION

One important class of computation is the extraction of **statistical properties** of input data. For our purposes, statistical properties are interesting regularities or features that may be made evident by suitable computation on the input data. Statistics are imprecise for a small number of data points, so it is essential to perform the computation over many inputs. All of us who have done laboratory experiments are familiar with at least some rudimentary forms of data analysis. Historically, people were taught to *take* the data in the laboratory by repeated experiments, and then to *analyze* the data later. There are situations, however, in which we cannot make up for bad data by doing more experiments. The people who study earthquakes do not get to go back and collect more data. They can wait for the next earthquake, but it will be a different earthquake—a different experiment. A number of events associated with natural disasters cannot be replicated—the Mount St. Helen's eruption, for example. We can study what remains, but we do not get to do the experiment again.

Our sensory systems are perhaps the best example of sources of data that cannot be repeated. We never see the same scene twice—even if we try. We perceive familiar objects in a different position, with a different background, each time. The same predator is jumping on us from a different tree—there is no time to take a second look. Those animals that could extract the most information from a fleeting glance were most likely to survive, and to pass on that processing capability to their progeny.

Whenever we cannot replicate the conditions and collect several batches of data, the data that we do get are sure to contain some *bad data points*. Perhaps the voltmeter changed ranges and did not catch up with itself in time to take the reading correctly. Perhaps you walked across the room and zapped the experiment. Or, perhaps a cosmic ray came along. In the nervous system, neurons are dying all the time. We get not only good inputs from our visual and hearing systems—we get a lot of spurious inputs as well. No single input can be trusted completely. The system is designed to compute the most useful or informative result possible, in spite of inputs that are totally out of range.

Statisticians have various criteria by which, if a data point is sufficiently out of line, its effect may be reduced. A common procedure is to develop some notion of reasonable behavior. In an experiment, we often have sound theoretical reasons to believe that the output should be some smooth function of some independent variable (the input). The transistor curves in Chapter 3 and the amplifier transfer curves in Chapter 5 are examples. In both cases, there is a voltage scale given

by $kT/(q\kappa)$. If we change the input less than this amount, we do
output to change abruptly. Hence, if we take several data points
$kT/(q\kappa)$ voltage interval, we have a great deal of *redundancy* in the
distance from a single data point to a smooth curve passing through
of other points in the neighborhood is relatively large, we should
out that maverick data point. Any such scheme relies on four import

1. We know the size of a **region of smoothness** within which,
damental reason, the data cannot change abruptly
2. Many data points are available within the region of smoothne
3. A method, consistent with the nature of the expected smooth
able for fitting a smooth function through the data points
4. Some method of estimating the average deviation of the da
smooth function is available

Once we have formulated a computation with these attributes,
it to identify unexpected data points. These may be "bad" points,
be items of exceptional interest. Isolated points do not exist in a ele
image. It is desirable, however, to see stars in the night sky. One co
bad datum is another computation's exceptional event. Sensory pro
plete with examples of spatially and temporally smoothed signals. T
functions are used to provide a reference for local computation. The
known example is the **center-surround** organization of many visual
retina to cortex. The signal average in a central area is subtracted
erage over a much larger surrounding area; the resulting difference
as the output. A similar organization is found, in some form, in all
sory pathways. In the following sections, we will discuss several ways
spatial averages of an ensemble of inputs.

FOLLOWER AGGREGATION

The simplest circuit for computing a smooth function is shown in
It consists of n follower stages, all driving the single wire labeled V
reaction to this **follower-aggregation circuit** might well be, "W
to make the V_{out} wire follow every input—and it obviously cannot
input. It is an n-way follower, but there can be only one output voltag
seen previously the importance of signal *types*. In the circuit of Fig
output of each individual amplifier is a *current*, whereas the output o
aggregation is a *voltage*; that voltage is the outcome of a collective
of the entire set of amplifiers.

There are n amplifiers, each responsible for the contribution of
to the common output. Each amplifier has a transconductance G_1
for A2, and so on to G_n for An. The Gs are set by the current
the transconductance amplifiers. We write Kirchhoff's law for the

ase the precision with which each individual input is treated. Appendix D s a formal treatment of several of the points we have mentioned here.

he balance of this chapter, we will examine examples of networks that e connectivity to compute interesting functions analogous to several of mputed in the brain.

ATISTICAL COMPUTATION

important class of computation is the extraction of **statistical prop-** f input data. For our purposes, statistical properties are interesting reg- or features that may be made evident by suitable computation on the ta. Statistics are imprecise for a small number of data points, so it is to perform the computation over many inputs. All of us who have done ry experiments are familiar with at least some rudimentary forms of lysis. Historically, people were taught to *take* the data in the labora- repeated experiments, and then to *analyze* the data later. There are s, however, in which we cannot make up for bad data by doing more ents. The people who study earthquakes do not get to go back and col- e data. They can wait for the next earthquake, but it will be a different ke—a different experiment. A number of events associated with natural cannot be replicated—the Mount St. Helen's eruption, for example. We y what remains, but we do not get to do the experiment again.

sensory systems are perhaps the best example of sources of data that e repeated. We never see the same scene twice—even if we try. We per- niliar objects in a different position, with a different background, each e same predator is jumping on us from a different tree—there is no time a second look. Those animals that could extract the most information eeting glance were most likely to survive, and to pass on that processing y to their progeny.

never we cannot replicate the conditions and collect several batches of e data that we do get are sure to contain some *bad data points.* Perhaps neter changed ranges and did not catch up with itself in time to take ing correctly. Perhaps you walked across the room and zapped the ex- . Or, perhaps a cosmic ray came along. In the nervous system, neurons all the time. We get not only good inputs from our visual and hearing —we get a lot of spurious inputs as well. No single input can be trusted ly. The system is designed to compute the most useful or informative ssible, in spite of inputs that are totally out of range.

sticians have various criteria by which, if a data point is sufficiently out s effect may be reduced. A common procedure is to develop some notion able behavior. In an experiment, we often have sound theoretical reasons e that the output should be some smooth function of some independent (the input). The transistor curves in Chapter 3 and the amplifier transfer Chapter 5 are examples. In both cases, there is a voltage scale given

by $kT/(q\kappa)$. If we change the input less than this amount, we do not expect the output to change abruptly. Hence, if we take several data points within each $kT/(q\kappa)$ voltage interval, we have a great deal of *redundancy* in the input. If the distance from a single data point to a smooth curve passing through the average of other points in the neighborhood is relatively large, we should certainly check out that maverick data point. Any such scheme relies on four important features:

1. We know the size of a **region of smoothness** within which, for some fundamental reason, the data cannot change abruptly
2. Many data points are available within the region of smoothness
3. A method, consistent with the nature of the expected smoothness, is available for fitting a smooth function through the data points
4. Some method of estimating the average deviation of the data from the smooth function is available

Once we have formulated a computation with these attributes, we can use it to identify unexpected data points. These may be "bad" points, or they may be items of exceptional interest. Isolated points do not exist in a close-up visual image. It is desirable, however, to see stars in the night sky. One computation's bad datum is another computation's exceptional event. Sensory processing is replete with examples of spatially and temporally smoothed signals. These smooth functions are used to provide a reference for local computation. The most widely known example is the **center-surround** organization of many visual areas, from retina to cortex. The signal average in a central area is subtracted from an average over a much larger surrounding area; the resulting difference is reported as the output. A similar organization is found, in some form, in all known sensory pathways. In the following sections, we will discuss several ways to compute spatial averages of an ensemble of inputs.

FOLLOWER AGGREGATION

The simplest circuit for computing a smooth function is shown in Figure 7.2. It consists of n follower stages, all driving the single wire labeled V_{out}. A typical reaction to this **follower-aggregation circuit** might well be, "We are trying to make the V_{out} wire follow every input—and it obviously *cannot* follow every input. It is an n-way follower, but there can be only one output voltage." We have seen previously the importance of signal *types.* In the circuit of Figure 7.2, the output of each individual amplifier is a *current*, whereas the output of the entire aggregation is a *voltage*; that voltage is the outcome of a collective interaction of the entire set of amplifiers.

There are n amplifiers, each responsible for the contribution of its V_i input to the common output. Each amplifier has a transconductance: G_1 for A1, G_2 for A2, and so on to G_n for An. The Gs are set by the current controls on the transconductance amplifiers. We write Kirchhoff's law for the node V_{out}.

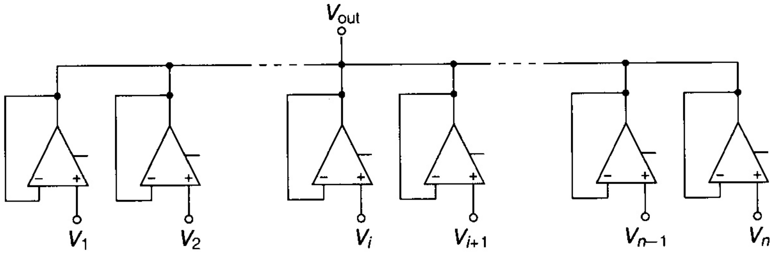


FIGURE 7.2 Schematic of the follower-aggregation circuit. Each follower supplies a current proportional to the difference between the input voltage and the output voltage. The contribution of each input to the output voltage is weighted by the transconductance of the associated amplifier.

The total current is the sum of the currents out of each amplifier. The current for the first amplifier is $G_1(V_1 - V_{\text{out}})$; that for the second is $G_2(V_2 - V_{\text{out}})$; that for the nth is $G_n(V_n - V_{\text{out}})$. Finally, the total has to be equal to zero:

$$\sum_{i=1}^{n} G_i(V_i - V_{\text{out}}) = 0$$

Transferring the V_{out} terms to the other side of the equation and rearranging, we obtain

$$V_{\text{out}} = \frac{\sum_{i=1}^{n} G_i V_i}{\sum_{i=1}^{n} G_i}$$

In other words, V_{out} is the average of the V_i inputs, each input weighted by its transconductance G_i.

Robustness

The follower-aggregation circuit computes the weighted average of the input voltages $V_1, \ldots, V_n$. Up to this point, our analysis has assumed a linear relation between input voltage and output current. This simplification has allowed us to write the solution, but has neglected what is probably the most charming attribute of the circuit and its relatives: The follower implementation of a neural network *has great* **robustness** *against bad data points.*

Transconductance amplifiers have a strictly limited current output. This limit is evident in their tanh transfer characteristics. The robustness of collective networks made with these circuits is a direct result of this current limitation. If any one input voltage is way off scale, it does not matter—the off-scale voltage will not pull any harder on the wire than would a voltage a few $kT/(q\kappa)$ different from the intended voltage of the wire. As long as all inputs are close to the average value, V_{out} will assume an average, with the inputs weighted by the current in their amplifier. The voltage V_{out} will not follow a few pathological inputs way off into left field.

Signals get stuck all the time, and biological wetware is even less relia
silicon hardware. We have 1000 to 10,000 inputs to a neuron—we can be s
some of them are going to be stuck on and some of them are going to b
off. If we have that many amplifiers with that kind of a fan-in, whether in
chip or in a neuron, some signals always will be stuck. There are just to
wires and too many amplifiers for all the components to be 100-percent

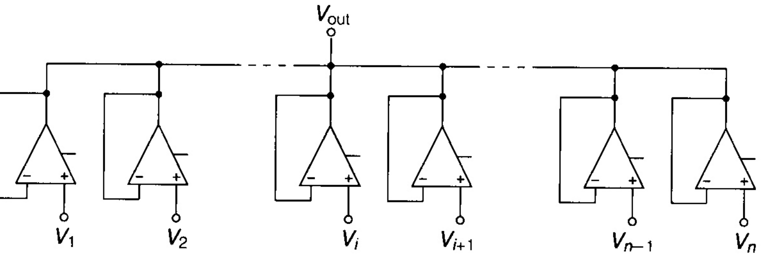
The follower-aggregation circuit shares with neural systems an exc
level of reliability in the face of failure of individual components. In bot
the robustness is a result of two factors: a large number of redundant
and a limited current that can result from any given input.

From a statistical viewpoint, the tanh characteristic changes the
tation done by the network. It implements what statisticians call a **re**
transformation: The weighting assigned to outlying data points is r
For all signals close to V_{out}, we have seen that the circuit computes a w
average, or *mean.* Signal values that are scattered by many $kT/(q\kappa)$ are
as inputs to a weighted *median* calculation. In both cases, the data are w
by the transconductances of their respective amplifiers. To ensure that
gle amplifier contributes more than its share to the output, we use wic
amplifiers to avoid the V_{min} problem described in Chapter 5.

RESISTIVE NETWORKS

The follower-aggregation circuit does a great job of computing an
that can be used as a reference against which to measure exceptional
There is a problem, however, with this kind of average. The average is repr
by the voltage on a single wire, and that wire is a single electrical no
average, therefore, will be a *global* average: It will extend physically to tl
remote location at which any input to it can originate. There are appl
for which this kind of global average is desirable. In most systems, howe
will need a much more *local* average, one in which the contribution of s
distant inputs is less than that of inputs in close proximity to the point a
the average is used.

An excellent example of local spatial averaging is found in the vis
tems of all higher animals. The illumination level within a visual scen
varies from one point to another by several orders of magnitude. If th
system used a global average as a reference, details in very bright and ve
areas would be invisible. A predator need only leap from the shadows—a
an arrangement would not make it into the next generation. For this
a locally weighted average signal level, from which local differences can l
sured, is computed by a layer of *horizontal cells* in the retina. These c
linked together by high-resistance connections called **gap junctions**, ar
an electrically continuous *resistive network* just below the photorecepto
horizontal network is by no means the only place in the nervous systen
this kind of a computation is used. As we mentioned in Chapter 4, the d
trees of all neurons are used to spread signals spatially, and the potentia



URE 7.2 Schematic of the follower-aggregation circuit. Each follower supplies a ent proportional to the difference between the input voltage and the output voltage. contribution of each input to the output voltage is weighted by the transconductance e associated amplifier.

al current is the sum of the currents out of each amplifier. The current first amplifier is $G_1(V_1 - V_{\text{out}})$; that for the second is $G_2(V_2 - V_{\text{out}})$; that nth is $G_n(V_n - V_{\text{out}})$. Finally, the total has to be equal to zero:

$$\sum_{i=1}^{n} G_i(V_i - V_{\text{out}}) = 0$$

ring the V_{out} terms to the other side of the equation and rearranging, in

$$V_{\text{out}} = \frac{\sum_{i=1}^{n} G_i V_i}{\sum_{i=1}^{n} G_i}$$

words, V_{out} is the average of the V_i inputs, each input weighted by its nductance G_i.

oustness

follower-aggregation circuit computes the weighted average of the input $V_1, \ldots, V_n$. Up to this point, our analysis has assumed a linear relation input voltage and output current. This simplification has allowed us the solution, but has neglected what is probably the most charming e of the circuit and its relatives: The follower implementation of a neural *has great* **robustness** *against bad data points.*

nsconductance amplifiers have a strictly limited current output. This vident in their tanh transfer characteristics. The robustness of collective s made with these circuits is a direct result of this current limitation. If input voltage is way off scale, it does not matter—the off-scale voltage pull any harder on the wire than would a voltage a few $kT/(q\kappa)$ different intended voltage of the wire. As long as all inputs are close to the average $_{\text{out}}$ will assume an average, with the inputs weighted by the current in plifier. The voltage V_{out} will not follow a few pathological inputs way eft field.

Signals get stuck all the time, and biological wetware is even less reliable than silicon hardware. We have 1000 to 10,000 inputs to a neuron—we can be sure that some of them are going to be stuck on and some of them are going to be stuck off. If we have that many amplifiers with that kind of a fan-in, whether in a VLSI chip or in a neuron, some signals always will be stuck. There are just too many wires and too many amplifiers for all the components to be 100-percent reliable.

The follower-aggregation circuit shares with neural systems an exceptional level of reliability in the face of failure of individual components. In both cases, the robustness is a result of two factors: a large number of redundant inputs, and a limited current that can result from any given input.

From a statistical viewpoint, the tanh characteristic changes the computation done by the network. It implements what statisticians call a **resistant transformation**: The weighting assigned to outlying data points is reduced. For all signals close to V_{out}, we have seen that the circuit computes a weighted average, or *mean*. Signal values that are scattered by many $kT/(q\kappa)$ are treated as inputs to a weighted *median* calculation. In both cases, the data are weighted by the transconductances of their respective amplifiers. To ensure that no single amplifier contributes more than its share to the output, we use wide-range amplifiers to avoid the V_{min} problem described in Chapter 5.

RESISTIVE NETWORKS

The follower-aggregation circuit does a great job of computing an average that can be used as a reference against which to measure exceptional events. There is a problem, however, with this kind of average. The average is represented by the voltage on a single wire, and that wire is a single electrical node. The average, therefore, will be a *global* average: It will extend physically to the most remote location at which any input to it can originate. There are applications for which this kind of global average is desirable. In most systems, however, we will need a much more *local* average, one in which the contribution of spatially distant inputs is less than that of inputs in close proximity to the point at which the average is used.

An excellent example of local spatial averaging is found in the visual systems of all higher animals. The illumination level within a visual scene often varies from one point to another by several orders of magnitude. If the visual system used a global average as a reference, details in very bright and very dark areas would be invisible. A predator need only leap from the shadows—and such an arrangement would not make it into the next generation. For this reason, a locally weighted average signal level, from which local differences can be measured, is computed by a layer of *horizontal cells* in the retina. These cells are linked together by high-resistance connections called **gap junctions**, and form an electrically continuous *resistive network* just below the photoreceptors. The horizontal network is by no means the only place in the nervous system where this kind of a computation is used. As we mentioned in Chapter 4, the dendritic trees of all neurons are used to spread signals spatially, and the potential at any

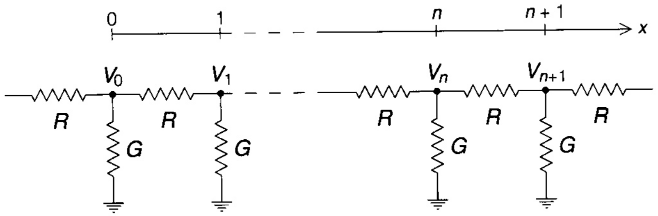


FIGURE 7.3 Resistive model of passive electrotonic spread in a neural process. The distance scale over which a signal dissipates is determined by the product *RG*.

given leaf of the tree is more affected by inputs in the immediate proximity than by those farther away. Propagation of signals in resistive networks is generically referred to as **electrotonic spread.**

Electrotonic Spread

The simplest example of electrotonic spread occurs in a long, straight, passive neural process of constant diameter. We can model the process as a resistive-ladder network, as shown in Figure 7.3. The R resistances correspond to the axial resistance per unit length of the cytoplasm, and the G conductances represent the leakage conductance per unit length through the membrane to the extracellular fluid. A potential V_0 is generated by an input at the left end of the process ($x = 0$). The voltage $V(x)$ generated by the input decreases with distance x from the input, because some of the current injected by the input is shunted to ground by the G conductances. We present a detailed analysis of continuous and discrete networks in one and two dimensions in Appendix C.

An important result, derived in Appendix C, is the rate at which signals die out with distance from the source. Intuitively, if the membrane-leakage conductance G is small compared with the conductance $1/R$ of the cytoplasm, the signal should propagate a large distance before it dies out. The greater the membrane conductance, the shorter the distance. For uniform, continuous networks, the voltage has the form

$$V = V_0 e^{-\alpha|x|} = V_0 e^{-\frac{1}{L}|x|} \tag{7.1}$$

where α is the **space constant** and L is the **characteristic length** or **diffusion length** of the process:

$$\alpha = \frac{1}{L} = \sqrt{RG} \tag{7.2}$$

A signal injected into a linear resistive ladder network decays exponentially with distance from the source. If a signal is injected into a node in the middle of a very long process, the influence of that input spreads out in both directions, not just in the $+x$ direction. For a one-dimensional model, the solution in the $-x$ direction is just the mirror image of the solution for the $+x$ direction. This observation is responsible for the absolute-value dependence on x in Equation 7.1.

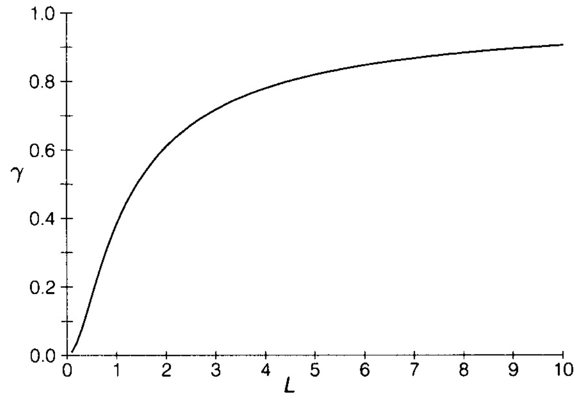


FIGURE 7.4 The exponent γ as a function of L, as computed from Equation 7.4.

For discrete networks, the decay also is exponential. For a node n sections away from the source, the voltage will be

$$V_n = \gamma^n V_0 \tag{7.3}$$

where

$$\gamma = \frac{V_1}{V_0} = 1 + \frac{1}{2L^2} - \frac{1}{L}\sqrt{1 + \frac{1}{4L^2}} \tag{7.4}$$

where $1/L$ is equal to $\sqrt{RG}$ as before, but in the discrete case the values of R and G are given per section rather than per unit length.

A plot of γ as a function of L is shown in Figure 7.4. For large values of L, γ approaches 1, and the continuous approximation of Equation 7.1 is valid. For values of L less than about 10, the magnitude of the decrement per stage given by the discrete solution differs markedly from that obtained from the continuous approximation. Later in this chapter, we will compare data from an experimental one-dimensional network with Equations 7.3 and 7.4.

A second important result derived in Appendix C is the **effective conductance** G_0 of the network. A voltage source V_0 driving one end of a semi-infinite network must supply a current $I = G_0 V_0$ into the network. From the point of view of a signal source, the network acts just like a single conductance G_0. For a continuous network, the effective conductance is given by

$$G_0 = \sqrt{\frac{G}{R}}$$

For discrete networks, the value of G_0 is somewhat different from that for continuous ones:

$$G_0 = \sqrt{\frac{G}{R}}\sqrt{1 + \frac{1}{4L^2}}$$

The G_0 values given here are for a terminated semi-infinite network. The effective

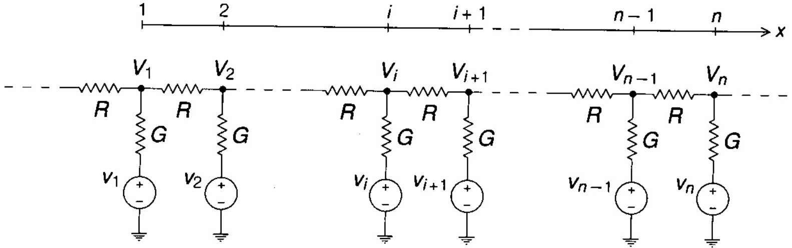


FIGURE 7.5 Electrotonic network in which input signals are supplied by voltage sources. This form of the network is equivalent to a current source in parallel with each G.

conductance at the center of a network that extends in both directions is $2G_0$, because a current V_0G_0 must be supplied for each direction.

Multiple Inputs

Multiple signal inputs to a network can be provided in the form of either voltage- or current-type signals. If we inject currents at many places, the network performs an automatic weighted average: the farther away the inputs are, the less weight they are given, in accordance with Equation 7.3. The voltage at any given point k due to a number of inputs is just

$$V_k = \frac{1}{2G_0}\sum_n \gamma^{|n-k|}I_n$$

In other words, the voltage at any point due to a number of inputs is just the sum of the voltages that would have been measured at that point had each input been presented individually, with all other inputs held at zero. This great simplification is a result of the principle of linear superposition mentioned in Chapter 2. The superposition result is true for any linear system, and is not dependent on the one-dimensional nature of the network, or on the fact that G and R were constant, independent of x. We will use it for two-dimensional networks, such as the horizontal network in the retina, and for our treatment of electrotonic spread in the dendritic tree. In these cases, the weighting function is more complex than the simple γ^n form of the one-dimensional network.

A convenient way to generate inputs to the network is to connect voltage sources in series with the conductances, as shown in Figure 7.5. Because the principle of superposition will hold for this arrangement as well, we need compute only the node voltage due to a single input. In Appendix C, we will derive the node voltage V_i generated by a voltage source v_i in the middle of a very long, uniform, discrete, one-dimensional network:

$$\frac{V_i}{v_i} = \frac{1}{\sqrt{4L^2+1}}$$

As the effective length over which the network averages increases, the effect of any given input decreases. For large characteristic lengths, the voltage due to any particular input is proportional to $1/L$. The total effect of a set of uniformly spaced inputs included in one characteristic length is therefore constant, independent of the value of L, because the number of inputs is proportional to L. This conclusion is clear if we observe that, when the voltage at all inputs is the same, the output voltage anywhere in the network is equal to the input voltage. We will make extensive use of both one- and two-dimensional networks with many inputs to derive local averages.

DENDRITIC TREES

Inputs to one neuron from the axons and dendrites of other neurons are aggregated by a tree of passive dendritic processes. For many years, scientists believed that the primary function of the dendrites was to collect input current into the main body, or **soma**, of the cell from which an action potential was generated. It is certainly true that neurons with axons do generate action potentials, or *nerve pulses*, as a result of current collected by their dendrites. Researchers have discovered in recent years, however, that the role of the dendritic tree is considerably more complex than was previously supposed [Shepherd, 1972; Shepherd, 1978].

Many types of neurons have no axon whatsoever, so their primary role cannot be to produce action potentials. Many types of neurons—those with axons and those without—have been shown to have *synaptic outputs* as well as inputs on their dendrites. This remarkable finding implies that much of the lateral communication in the nervous system is extremely local, and is mediated by graded analog (electrotonic) potentials rather than by the more digital nerve pulses. The dendrites convey two-way information rather than merely collecting current into the soma. Sorting out the far-reaching implications of these findings will require many years.

Synaptic Inputs

If enough current is injected into the dendritic tree, then the neuron will release neurotransmitter from any output synapses it has on its dendrites. If the current into the cell as a whole reaches a high enough level, the nerve can initiate pulses in its axon (if it has one). *Depolarizing* inputs cause the release of neurotransmitter from dendritic synapses and, if sufficiently intense and prolonged, can cause the axon to fire as well. These inputs are called *excitatory*. Inputs that *hyperpolarize* the neuron act to cancel out the effect of excitatory inputs; they are therefore called *inhibitory*.

If the entire path from the leaves of the dendrites to the axon hillock is less than L in length, the neuron is said to be **electrically compact**. Such a cell can be assumed to be equipotential throughout its dendrites, and therefore can be modeled as a wire. A neuron with dendritic processes much longer than

L can have very different potentials at different locations in its dendritic tree. The dendrites of such a neuron can be modeled as linear resistive networks. We will derive the voltage–current relationships at branches in a tree network in Appendix C.

Shunting Inhibition

We have used the voltage sources of Figure 7.5 to model excitatory and inhibitory input synapses to the network. Inputs also may be injected as currents, of course, one sign of current being excitatory and the other inhibitory. There is a third class of inputs, often called **veto synapses**, that neither hyperpolarize nor depolarize the neuron, but instead partially short-circuit to ground any activity present in the process. This kind of inhibition is called **shunting inhibition**.

The simplest realization of shunting inhibition is implemented directly by the network of Figure 7.5; we merely make one conductance, G_{shunt}, very large compared with the others. This arrangement will attenuate a signal traveling in either direction in the process. We will derive the attenuation suffered by a signal as it passes such a shunt in Appendix C. The result is

$$V_{\text{out}} = \frac{V_0}{1 + \frac{G_{\text{shunt}}}{2G_0}}$$

where V_0 is the voltage that would have been present without the shunt. As G_{shunt} becomes large compared with the network effective conductance G_0, the operation performed by such a synapse resembles a division by G_{shunt}.

Shunting is one of those wonderful biological tricks by which an input can inhibit activity, but not create any activity of its own. In a complex biological system, it often is difficult to distinguish shunting inhibition from inhibitory synapses that contribute a net negative current to the tree. The same basic synaptic mechanisms are used in both cases. Release of neurotransmitter causes channels to open in the postsynaptic membrane. An inhibitory synapse causes an increase in conductance for an ion with a negative reversal potential. A shunting synapse causes an increase in conductance for an ion with a reversal potential near the resting potential of the membrane. From a system perspective, however, it makes a world of difference whether the operation is a subtraction or a division. For this reason, we must exercise care when we use the biological literature as a basis for electronic models. Biological distinctions that seem insignificant at the descriptive level may have profound effects on the performance characteristics of the neurobiological system.

TWO-DIMENSIONAL NETWORKS

The horizontal network in the retina is a flat mesh of dense processes that are highly interconnected by resistive gap junctions. These interconnections are somewhat random in number and direction. Any given cell is connected with many others, and there is a great deal of overlap among interconnected cells. In silicon, discrete two-dimensional networks are very useful, and generally are implemented in a regular array by interconnection of nearest neighbors. We have mentioned that this kind of network computes an average that is a nearly ideal way to derive a reference with which local signals can be compared. In Chapter 15, we describe a retina containing a hexagonal two-dimensional network, with six resistors coming into each node, as shown in Figure 7.6. A resistance R is connected between neighboring nodes, and a conductance G (not shown) is connected from each node to ground (as in Figure 7.3; in Figure 7.5, it is connected to a voltage source).

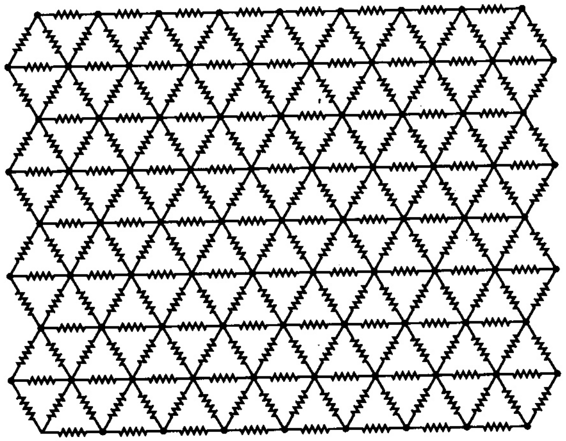


FIGURE 7.6 Topology of a hexagonal network. Because of its high degree of symmetry and redundancy, this network is the preferred form for two-dimensional applications.

This network is particularly attractive, because it has the highest symmetry and connectivity of any regular, two-dimensional structure. If we inject a current into a node of the network (which we will call node 0), the resulting voltage decays exponentially with distance from that node. We can derive an approximate solution for the decay law in the following manner. As we progress outward from node 0 following a row of resistors, we encounter nodes that are vertices of larger and larger hexagons centered on node 0. The index of hexagon n (its "radius") is just the number of resistors we must pass through on the direct path from node 0 to a vertex. Our **circular approximation** assumes that all nodes on the perimeter of a given hexagon have the same voltage. Under this approximation, we can write a finite-difference equation for the current into hexagon n in terms of the voltage relative to that of hexagon $n-1$ and to that of hexagon $n+1$. We notice that there are $6n$ nodes on the perimeter of hexagon n, and that there are $12n-6$ resistors from hexagon $n-1$ to hexagon n, and $12n+6$ resistors from hexagon n to hexagon $n+1$. The current I into hexagon n is therefore

$$I = \frac{(12n-6)(V_{n-1}-V_n) - (12n+6)(V_n - V_{n+1})}{R} - 6nGV_n \qquad (7.5)$$

Hoeneisen, B. and Mead, C.A. Current–voltage characteristics of small size MOS transistors. *IEEE Transactions of Electron Devices*, 19:382, 1972b.

Jacoboni, C., Canali, C., Ottaviani, G., and Quaranta, A.A. A review of some charge transport properties of silicon. *Solid State Electronics*, 20:77, 1977.

Maher, M.A. and Mead, C. A physical charge-controlled model for MOS transistors. In Losleben, P. (ed), *Advanced Research in VLSI.* Cambridge, MA: MIT Press, 1987, p. 211.

Shockley, W. *Electrons and Holes in Semiconductors, With Applications to Transistor Electronics.* New York: Van Nostrand, 1950.

Tsividis, Y.P. *Operation and Modeling of the MOS Transistor.* New York: McGraw-Hill, 1987.

APPENDIX C

RESISTIVE NETWORKS

Resistive networks play a central role in level normalization in a great many neural systems; an example of such a network was presented in Chapter 15. Analyses of such networks are scattered throughout the literature, often couched in a terminology that is specific to another discipline. For this reason, it is desirable to gather the relevant material in a single place, using a consistent notation.

ONE-DIMENSIONAL CONTINUOUS NETWORKS

The simplest resistive network is shown in Figure 7.3 (p. 108). It has a longitudinal resistance R per unit length, and a conductance to ground G per unit length. A potential V_0 is applied to the left end of the process ($x = 0$). The network is assumed to be semi-infinite, and R and G are assumed to be independent of x. We can determine how the input affects the voltage $V(x)$ on a node of the network at some value of x by writing the relations between the voltage and the current $I(x)$ flowing through the resistance R at that value of x, and of the same variables slightly farther along the line at $x + dx$:

$$V(x) = V(x + dx) + I(x + dx)R\,dx \tag{C.1}$$

$$I(x) = I(x + dx) + V(x)G\,dx \tag{C.2}$$

In the limit where dx becomes very small, I and V become continuous functions of x. Equation C.1 then becomes

$$\frac{dV}{dx} = -IR \qquad \text{(C.3)}$$

and Equation C.2 becomes

$$\frac{dI}{dx} = -VG \qquad \text{(C.4)}$$

Differentiating Equation C.3 with respect to x, and substituting Equation C.4 in the right-hand side, we obtain a second-order differential equation for V:

$$\frac{d^2V}{dx^2} = -R\frac{dI}{dx} = RGV \qquad \text{(C.5)}$$

R and G are constant, so the solution to Equation C.5 has the form

$$V = V_0 e^{-\alpha x} = V_0 e^{-\frac{x}{L}} \qquad \text{(C.6)}$$

We have ignored the $e^{\alpha x}$ solution because it diverges for large x. That solution will be appropriate, of course, for a network running in the $-x$ direction. The constant α is the *space constant* and L is the *characteristic length* or *diffusion length* of the network:

$$\alpha = \frac{1}{L} = \sqrt{RG} \qquad \text{(C.7)}$$

As we noted in Chapter 6, a signal can be represented either by a voltage or by a current. If the signal is a current, it can be injected directly into a node of the network. We can determine the magnitude of the injected current required at $x = 0$ to produce the voltage V_0 by substituting Equation C.6 into Equation C.3:

$$I(0) = V_0\frac{\alpha}{R} = V_0\sqrt{\frac{G}{R}} = V_0 G_0 \qquad \text{(C.8)}$$

The value $G_0 = \sqrt{G/R}$ is the **effective conductance** of the semi-infinite network. From the point of view of a signal source, the network acts just like a single conductance G_0.

If a signal is injected into a node in the middle of a very long process, the influence of that current spreads out in both directions, not just in the $+x$ direction. For a one-dimensional model, the solution for the $-x$ direction is just the mirror image of the solution for the $+x$ direction. For that reason, the effective conductance at a node of a network that extends in both directions is $2G_0$, because a current V_0G_0 must be supplied for each direction.

DISCRETE NETWORKS

The results of the previous section are valid for only continuous or very nearly continuous networks—those for which RG is much less than 1. For most applications, this restriction is not satisfied, and we must treat the general case where R and G can take on any values. We will begin by deriving a finite-difference equation for the voltage V_n on the nth node of the discrete line of Figure 7.3 (p. 108). The situation at one section of the line is shown in Figure C.1. We can derive the exact behavior of the discrete network by writing the circuit relations for two adjacent sections.

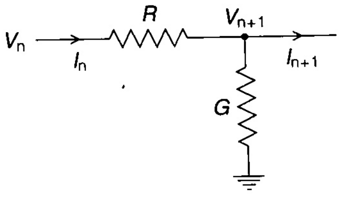


FIGURE C.1 Schematic showing relations between the voltages and currents in one section of a semi-infinite resistive network.

First, we express the current through the resistance R, connected between node n and node $n+1$, in terms of the voltages on those two nodes:

$$I_nR = V_n - V_{n+1} \qquad \text{(C.9)}$$

The same relation holds for the second section:

$$I_{n+1}R = V_{n+1} - V_{n+2} \qquad \text{(C.10)}$$

The current through the conductance G is just the difference between I_n and I_{n+1}:

$$GV_{n+1} = I_n - I_{n+1} \qquad \text{(C.11)}$$

Equation C.11 assumes that no current is injected directly into the node from an external source; in other words, we are looking for the natural spatial response of the network when current is injected into a single node. The effects of this excitation die out as we move away from the point $n = 0$ where the current is injected.

Substituting Equations C.9 and C.10 into Equation C.11, we obtain

$$GRV_{n+1} = V_n - V_{n+1} - V_{n+1} + V_{n+2}$$

Simplifying, we obtain the second-order finite-difference equation that the voltages of the nodes must satisfy:

$$V_{n+2} - (2 + RG)V_{n+1} + V_n = 0 \qquad \text{(C.12)}$$

We expect a solution that dies out exponentially as we move away from the source. We construct a trial solution of the form

$$V_n = \gamma^{|n|}V_0 \qquad \text{(C.13)}$$

Substituting this form for V_n in Equation C.12, and assuming $n > 0$, we obtain

$$V_0\gamma^{n+2} - (2 + RG)V_0\gamma^{n+1} + V_0\gamma^n = 0$$

Dividing by $V_0\gamma^n$, we obtain

$$\gamma^2 - (2 + RG)\gamma + 1 = 0 \qquad \text{(C.14)}$$