

Chapter 3

Neurons and Circuits

The fundamental computing unit of the brain is the neuron. Owing to extensive structure it has complex computing capabilities. You should think of it as a kind of microprocessor that is optimized for computing special functions. Many details about the way the neuron processes its inputs and produces an output are known, but nonetheless we still have an incomplete picture of all the neuron's capabilities.

3.1 Neurons as special cells

“The brain uses stereotyped electrical signals to process all the information it receives and analyzes. The signals are symbols that do not resemble in any way the external world they represent, and it is therefore an essential task to decode their significance.[4]”

Like any organ in the body, the brain is composed of cells. But the brain's cells, termed nerve cells or *neurons* are geometrically unlike any other cells in the body and are specially designed to send electrical signals over long distances. As Allman emphasizes[1], this was a technological breakthrough that allowed sensory organs to direct distant motor cells.

These cells form individual circuits that are specialized for each of the brain's different subsystems. The problem of figuring out what they do is complex and our comprehension at this point is still full of holes even though enormous amounts of information has been learned. To understand the magnitude of the problem one can start with the issues of levels of abstraction. A neuron is characterized by a very complex chemistry that governs all aspects of its functioning. Molecules internal to the cell can change its electrical properties and thus make it function differently. For the most part we will gloss over this level of detail and assume that the neurons have invariant properties during the timescales of interest, 1 millisecond to 1 second. The second level of detail is the electrical functioning of the neuron itself which is also complicated and depends on a number of different ionic channels that pump charge in and out of the cell.

Here again we will oversimplify matters and resort to a basic model wherein inputs add or subtract charge to the cell multiplicatively through electrical contacts. And finally there is the complexity introduced by having the neurons connected to each other in circuits. Such circuits are obviously doing many different kinds of functions, but our plan is just to sample two of these to convey the general idea.

A nerve cell has three main components, its dendrites, its soma or body, and its axon. The dendrites are a tree-like structure that form the main area where axons of other cells make connections. The connections are anatomically specialized structures termed synapses (See Figure 3.1). Neurons are unlike other cells in many ways but ones of the chief differences is that they are especially adapted for propagating electrical signals over distances. These distances are enormous when measured by the standard of the size of a standard cell. The body of the cell, its *soma*, is about ten microns(millionths of a meter) across, but if you imagined it to be the size of a marble, then its axon would extend for 10 meters or about thirty feet. Furthermore the axon bifurcates repeatedly until it has approximately 10,000 terminals that connect to other neurons at synapses.

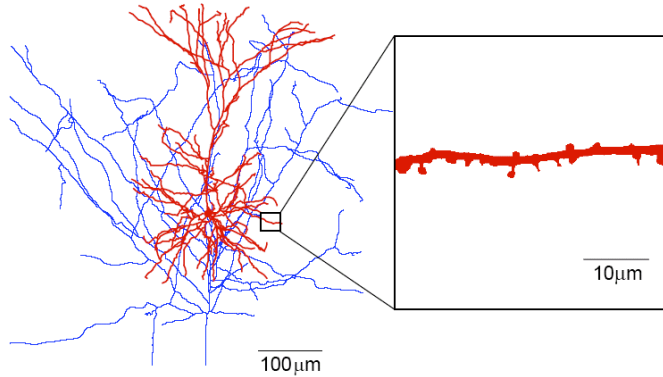


Figure 3.1: When the axon arrives at a target cell it encounters one of the cells extruded processes - a dendrite- where it makes a connection. To help things along the connection is made at a bouton, a micro-extrusion that provides a site for the connection. As shown, the collection of dendrites is peppered with synaptic boutons, to accommodate the thousands of incoming signals.[Permission Pending]

If you think of the axon as a small "wire," then you have so much such wire in your head that if you attached it end on end there would be enough to get to the moon and back. Clearly its the connections between cells that are key.

The total number of neurons in your brain is estimated to be on the order

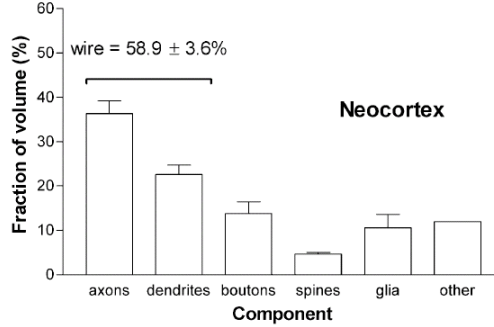


Figure 3.2: The brain is mostly wire.

of 10^{11} . Taking this number together with the figure of 10,000 connections per cell gives us 10^{15} connections total. While the nerve cells working together form a very complex system, the connections are a major, if not *the* major component that describes how the brain will process information. So much so, that scientists working on large scale models of neural networks are characterized as *connectionists*.

3.2 Signaling

Nerve cells in the brain communicate via voltage spikes. This is a complicated process that was worth a Nobel prize for Hodgkin and Huxley who described the first generally useful model[2], but for the moment we will present it in caricature. Each axon connecting to another cell's dendrite via a synapse is able to cause electrical charge to build up on the target cell. When some characteristic of that charge is satisfied, the target cell will produce a voltage pulse that has a typical amplitude of about 5 millivolts and a pulse width of about one millisecond. that pulse travels down the target cell's axon at about 10 meters per second. When it comes to axonal branches, the pulse is divided and proceeds along each branch. The special construction of the axon allows it to propagate. You could not do this with plain electrical wire as the capacitance of the wire would attenuate the signal too drastically, but biology does not have this problem as it is specially designed to actively propagate the spike. The speed of 10 meters per second might not sound like much, but when you consider that most of the brain is reachable in 10 centimeters, it will only take a millisecond to get the pulse to any particular new target neuron. Once it gets there, it deposits its charge and the process repeats. For a long time it was thought that this process of dumping charge on the target neuron was deterministic but some recent research suggests that it may be a probabilistic process.

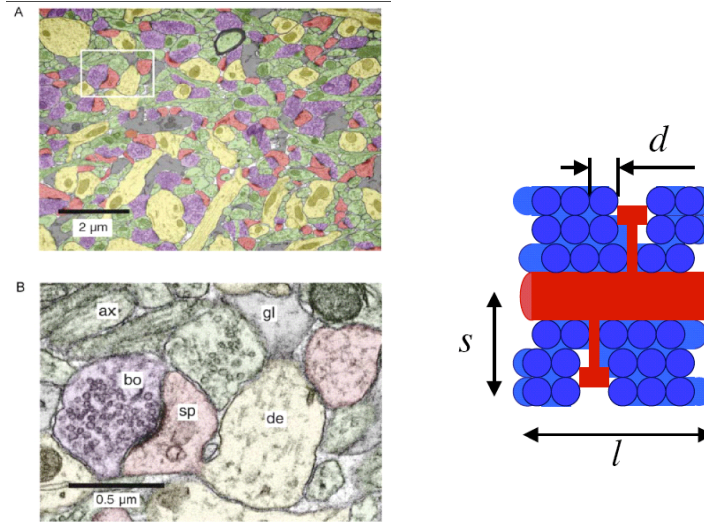


Figure 3.3: (*left*) Two electron micrograph images at different scales show dramatically how packed the brain is in terms of its neural circuitry. The micrographs have been colorized to denote different neural components. (*right*) Formal models by Chlovskii that analyze the efficiency of connections between cells have suggested that the packing is optimal.[Permission pending]

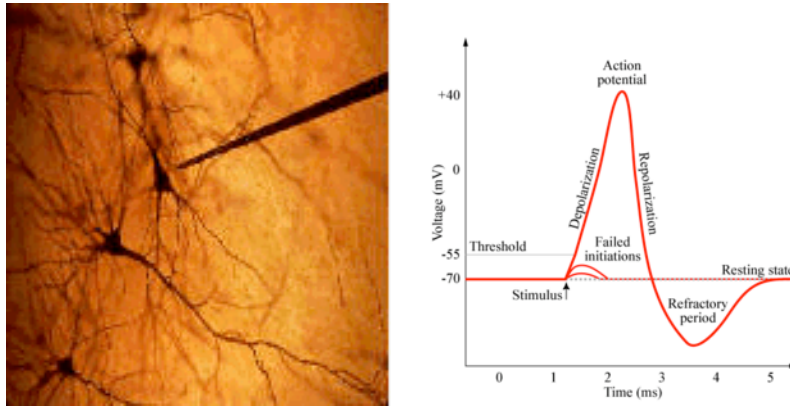


Figure 3.4: (*left*) Single neuron recording techniques allow the measurement of the basic signaling method of a neuron, the action potential. If a small wire can be advanced sufficiently near the soma of a neuron, the voltage changes in the cell body can be measured. (*right*) A schematized version of the voltage spike or action potential created by adding charge to a neuron. At threshold the soma suddenly depolarizes or increases in voltage and then just as suddenly reverses itself.[Permission pending]

3.3 Signaling

Assume that the pulses encode a number somehow in the form of voltage charge. Then at any moment the target cell is accumulating charge that will determine whether or not it sends its own pulse. A popular model is termed “integrate and fire,” meaning that the cell simply adds up its incoming charge and sends its pulse when the sum crosses some threshold. Newer models are more complicated and supercede the integrate-and-fire model but nonetheless this model will be a useful thinking tool with which to describe other essential neuron features.

At the time scale of 100 milliseconds to one second, the action potential just looks like a single vertical line on an oscilloscope, so that it is compelling to think of the signal at this level as a binary code where a spike means a ‘one’ and no spike means ‘zero.’ The sequence of ones and zeros over time is a code used by all spiking neurons to communicate. Although as Collins et al note in the quotation at the beginning of this chapter, the sequence represents a compact code, it has not yet been cracked. Nonetheless for certain neurons simple abstractions convey a lot of useful information. One such abstraction is the idea of a rate code. This idea is enormously compelling and has been a hallmark of thousands of neural recording experiments. Figure 3.5 shows a typical observation from a sensory neuron. This one is taken from an area in the cortex that is measuring motion. In a smallish area of visual space. if the world is moving in a certain direction. the neuron signals this by sending lots of spikes. Whereas if the world is moving in the opposite direction, the neuron sends just a few spikes. The fastest that neurons can spike is about 500-1000 spikes per second, but most forebrain neurons, and especially cortical neurons are much slower, spiking at rates of about 10 spikes per second.

Given its status as a ubiquitous observation, why should there be any doubts about the use of a rate code as the fundamental message protocol between neurons? There are several reasons to be skeptical but the main one is that the rate code is a unary code. To send a larger number there must be more spikes and the number of spikes is proportional to the number’s magnitude. Note that this is very much more expensive than a binary code used by silicon computers where the number of binary bits sent is proportional to the logarithm of the number that needs to be sent. This difference is very important as the main expense for a neuron is sending a spike[3]. And the brain itself is metabolically expensive. It only weighs 2% of total body weight but consumes approximately 20% of the metabolism.

Because sending a unary-code spike is so expensive researchers have been exploring alternate coding strategies that would use the spikes more efficiently but so far no alternate scheme has been found. Numerous recent experiments show that the neurons exhibit many different kinds of oscillations however so that oscillation-based codes may ultimately replace the rate-code model.

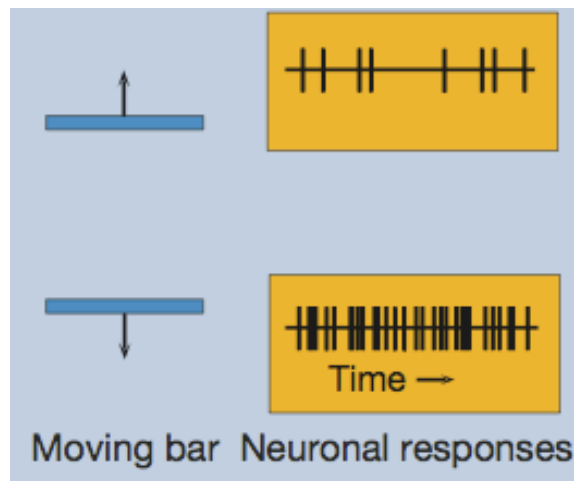


Figure 3.5: A visual motion sensitive cell in the cortex studied by Newsome. Motion at a certain location in space in a preferred direction produces a large amount of spikes, in contrast to motion in the opposite direction which produces few spikes [Permission pending].

3.4 Synapses

The basic model of a synapse is simple in that it provides an electrical connection between cells. However the chemical machinery that creates, maintains and manages it is very complex. As you know now, the axons of a cell may have to go on extraordinary journeys to find their targets. They do this during a developmental period by following chemical gradients like bloodhounds following a scent. They can use a mixture of molecules for complicated steering and then stop when they identify markers on the surface of a target cell. Once they stop, there are specialized *glial cells* to manufacture a synapse. For a long time glial cells were an embarrassment to neuroscientists who did not have much of an idea as to what they did. The embarrassment arose because of two reasons. First there were a lot of glial cells - about one to ten times as much as the rest - and second the glial cells did not produce voltage spikes. Since it is widely believed that such spikes are the essence of brain signaling and the genesis of thought, it therefore seemed that the glial cells were a sidebar to the thinking process.

Now there is significant evidence to suggest what glial cells do and that is make synapses. They do this by using a combination of a special protein together with a measure of the electrical activity between the incoming axon and its potential target cell. Its a bit of a chicken and egg problem but if the electrical activity of the two are related, then the glial cell can build a more permanent connection between them in the form of a synapse. Synapses can be of two forms: *excitatory* where charge is added or *inhibitory* where charge is

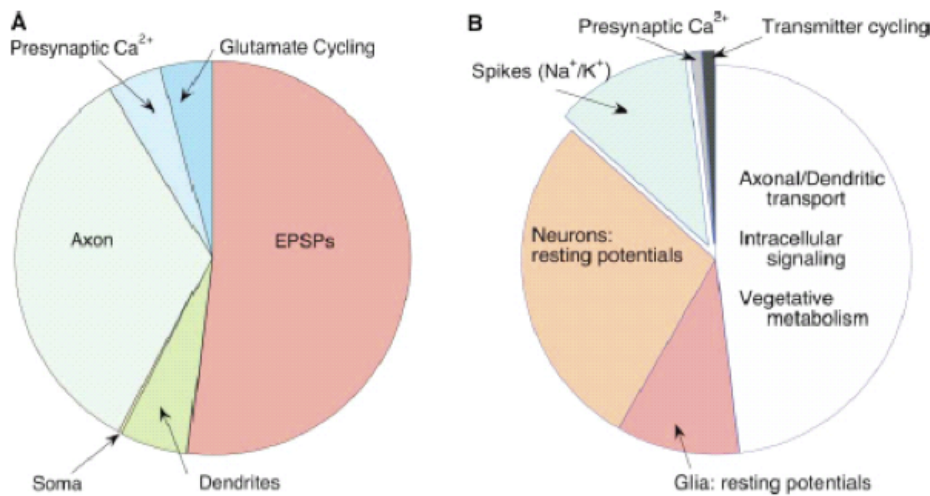


Figure 3.6: Sending a spike is very expensive for a cell[3]. On the left we see that the excitatory post synaptic potentials - read spikes - consume more than half a neuron's energy. [Permission pending].

removed. A single neuron can only have synapses of one or the other type.

3.5 Receptive fields and neurons as codes

Now that we've covered neuron basics, its time to put them to use and give our cells something to do. One central neuronal job is that of coding the external world into symbols, so we'll start here and see how that could be done. The model will be quite compelling so we'll have to resist the urge to over-generalize it, and in fact it will need many revisions later. But it contains so many useful ideas that will stand us in good stead in discussions ahead, that we'll take it very seriously here. The examples will be taken from studies of vision in monkeys. The reasons are that vision has been the most studied and that monkeys have neural wiring that is most like ourselves.

To make the idea of encoding concrete, consider how the eye handles an image. Light impinging on the eye contacts a matrix of discrete photoreceptors. These are specialized in terms of their wavelength sensitivities. *Rods* respond to all wavelengths of visible light and are very sensitive to low light levels. *Cones* come in three wavelength sensitivities peaked at red, green and blue respectively. Cones are concentrated near the center of the optical axis and become sparser towards the periphery where rods are dominant. We can see color in the periphery, but the spatial resolution of that 'image' is drastically reduced.

After the retina, the image is sent to the thalamus, specifically to a small

portion of the thalamus called the lateral geniculate nucleus or LGN. The LGN is further distinguished as the place where inputs from both eyes are proximal. There are two LGNs and they each look after a hemifield of visual space. Looking down from the top, the left LGN takes care of inputs from the space to the right side of vertical. We know for a fact that the signal that arrives here is heavily coded since the amount of cells sending axons to the LGN is about 100 times less than the rods and cones detecting light initially. However the exact form of the transmitted code has yet to be determined in detail. The code used by the LGN is known in broad outline though. It is in the form of cells that have a responsive center and an inhibitory surround, as shown by Figure 3.7.

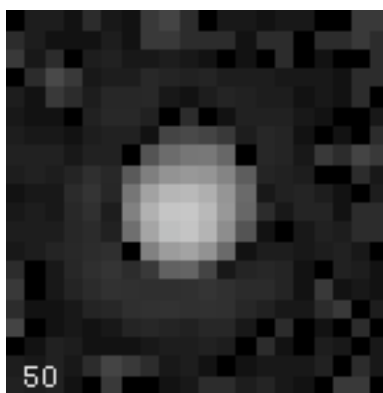


Figure 3.7: The high resolution image of the retina is summarized by more coarsely sampled cells in the lateral geniculate nucleus(LGN) of the thalamus. The cells that code for light intensity are of two principle kinds: those that respond best to a lighter-than-average center spot of light- ON cells - and those that respond to a darker-than-average center spot of light - OFF cells. The figure shows an ON cell.[Permission pending].

You can broadly think of the dots in a conventional dot matrix printer, except that there are white dots and colored dots as well as black dots. You might think that a black dot could be signaled as the negative of a white dot, but the brain has apparently not invented explicit negative numbers and instead resorts to a push-pull representation: For every value signaled, there is a cell dedicated to its logical opposite. By peculiar convention cells that prefer a lighter center are termed ON cells and those that prefer a darker center are termed OFF cells but don't allow this idiosyncratic naming convention to confuse you. Both types have the same behavior in that they fire when the pattern they prefer occurs.

3.6 Coding Sensory Data

The coding at the LGN is just the beginning of an elaborate process that is handled by the cortex. What starts out as a collection of light and dark dots

ends up as neurons that are responsive to the emotional expressions of faces. We will have more to say about how this process operates in the next chapter. For the moment we will just follow it for one more step to make a number of points about neural circuitry and its models. What we are calling the dot matrix representation is unsatisfactory from the perspective of a code because it contains the ability to code many images that never are generated by the world with any significant probability. Pause for a moment and take a look at your surroundings. What you are likely to notice is a scene of textured surfaces circumscribed by sharp boundaries where the intensity changes quickly over a small spatial extent. What you won't see is anything that looks even vaguely like television "snow." However the LGN neurons potentially can code the scenes from the real world and snow with equal facility. But since television snow almost never comes up for any significant interval, and contains no useful information, a system that can code it equally is a waste.

A coding scheme geared to the world would be especially organized to selectively encode images that are more likely to happen and discount those images that are very unlikely to occur(We say "unlikely since once and a while we do look at television "snow."- even though we might have better things to do.) Now when we look at the connections from the LGN to the visual cortex, the first stage of the memory, they seem to be doing just this: picking a coding scheme that emphasizes the structure in naturally occurring images at the expense of random noise.

Now we'll describe an encoding scheme that does the job for encoding small image patches. To do this we need the idea of a set of *basis* images. One such image is shown in Fig 3.8. Suppose you would like to make a neuron that was responsive to such images. How would you go about it? Take a minute to think about this problem.

Hopefully what might be occurring to you is that you could do it in a straightforward way by using the neurons in the LGN. Where you want your neuron to be responsive to a relatively light pattern, you would find an ON cell at that location and connect it to your cell. You would do the same for the darker areas using OFF cells. As shown in Figure 3.8, you can do this by taking the axons of the requisite cells and connecting them with synapses to the coding cell. To emphasize a particular value at a particular point you can make the strength of the connection large. The simplest model used multiplies the input by strength of the connection to obtain its overall effect. Thus when the pattern appears in the dot coding cells it has the maximal effect on the coding cell.

Note that you cannot solve this problem in the obvious way using negative numbers because there are not any! If you had them it would be easy. Number the synapses \mathbf{w} sequentially from 1 to n . If the image is dark, just use a negative value. The to get the response r of the neuron, multiply the image \mathbf{x} by its synapses as follows:

$$r = x_1w_1 + x_2w_2 + \dots x_nw_n$$

You can see when the image \mathbf{x} is a scaled version of the synapses the response will be highest. This still works with the positive-synapse convention since the

‘negative’ x is in fact positive - since it is the output of an OFF cell and it pairs up with an excitatory synapse counterpart. Thus looking at the connections, the only way we would know to draw the connections in the light and dark format is to know where they came from. This is, very famously, the idea of the *labeled line*.

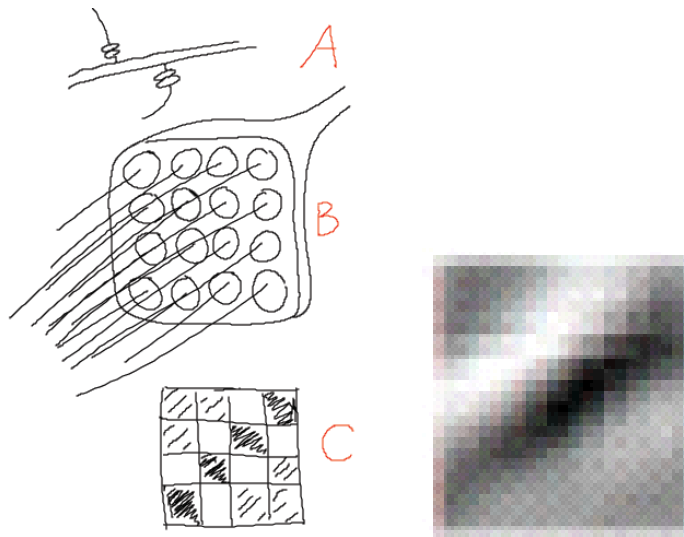


Figure 3.8: How to make a neuron that is sensitive to a particular pattern. The inputs that you need are connected to neurons on the neuron’s dendrite. If some of them are active the neuron may fire, but is maximally responsive to the presence of the entire pattern at its input.

Once you understand how one of these is created, the next step is to see how a *set* of them could be used. A principled way to do this would be to have the set of pattern neurons be related to the image in a systematic way. Figure 3.9 shows a portion of such a set.

Imagine that a general image is present and that these pattern neurons have to signal its presence. The trick is to weight each pattern appropriately and add them up. It turns out that for almost all natural images, it is possible to choose the weights to approximate the original image. The neurons’ weights in such a representation can be described mathematically as *basis functions* and their corresponding responses as *coefficients*. Here again you need the concept of the labeled line. The coding neurons send their information on in terms of the responses. The particular axons, or labeled lines, denote the source or meaning of associated with the particular inputs.

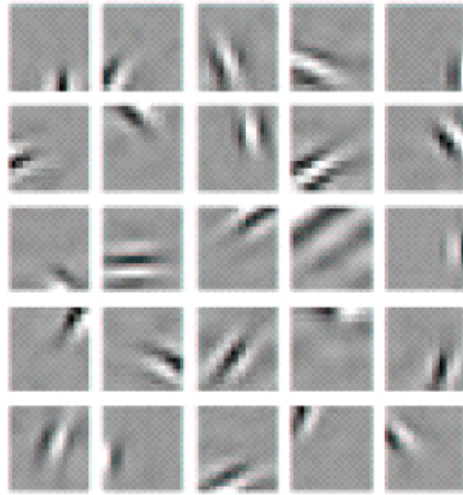


Figure 3.9: A portion of a set of basis functions obtained by Olshausen and Field's numerical simulations.

3.7 Circuits

At this point you have been exposed to the feed forward analysis of neural codes. Starting from the retina, information can be abstracted from the 'raw' sensory data in successive stages. Only two - the LGN and the first stage of visual cortex V1 have been discussed but later stages embody the same kind of thinking. It is just that the basis functions become more and more abstract and relevant to behavioral goals.

However we have not discussed how the receptive fields get coded in the first place. For this to happen there must be a normal developmental path but also experience with natural images. Furthermore it is very likely that this experience is used with feedback circuits to set up the appropriate synapses. Cortical circuits make extensive use of feedback with the feedback path typically having ten times the connections as the feed forward path. The first stage of circuitry from the LGN and the cortex is no exception. Figure ?? shows the basic connections. The lower left inset shows a detail that we have not discussed so far. The left LGN integrates the connections from the right side of both the left and right eyes. Furthermore there are two principle kinds of cells, *magnocells* that have coarse spatial information and relatively good temporal information, and *parvocells* that signal relatively good spatial information and relatively poor temporal information. There are two layers of parvocells for each eye and one layer of magnocells for six layers in all. All these cells send their connections to the input layer of V1, layer IV. In return V1 send feedback connections to the LGN from its upper and lower layers.

How can the feedback circuit help to setup the receptive fields of the V1

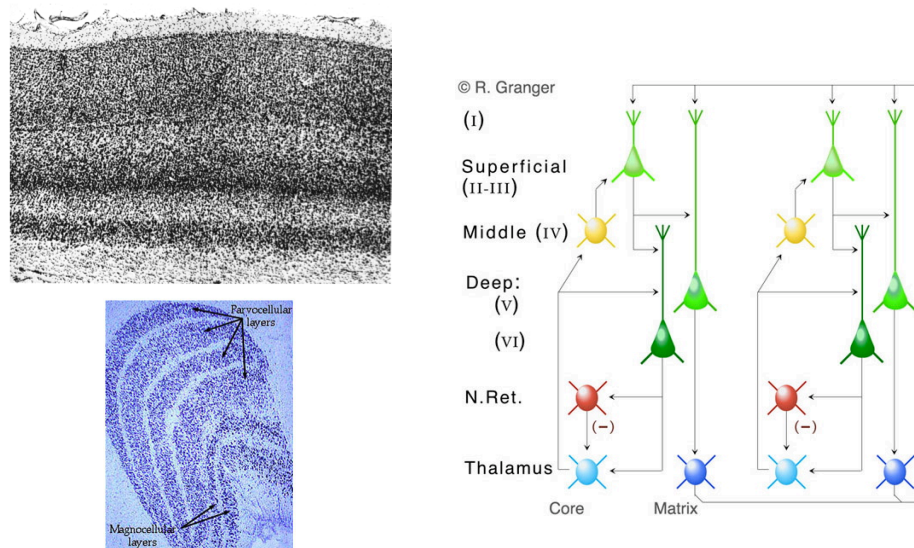


Figure 3.10: The distinct brain subsystems all have characteristic circuits. This example shows the circuitry that connects the cortex to the thalamus. The cortex has a six layered structure that consists of an input layer (IV) output layers (V and VI) and processing layers (I - III). The output layers feed back to the thalamus that are ten times as numerous as the input connections from the thalamus to cortical layer IV. The rightmost part of the figure shows a schematization of the connections by Granger. In this diagram cortical cells are green and yellow and LGN cells are blue.[Permission pending:Granger].

cells? This process can be described in terms of a learning algorithm wherein images are applied to a model retina, the signals are transformed to simulate a set of LGN responses and then an algorithm is used to adjust the synapses of V1 cells. To explain the algorithm we'll start with a version that is almost right and then refine it. The algorithm uses concepts from geometry and linear algebra, so if you need it, the following box is a primer. If you don't just skip over it.

IMAGES AND VECTORS

Much of the thinking about neurons and their receptive fields rests on geometric concepts. For instance an image is an array of brightness values. To acknowledge the discrete nature of this representation, the image samples that they represent are called *pixels* and the associated values 'pixel values.' Below shows

an example of a small sixteen pixel image.

3	7	15	18
6	1	20	22
2	5	8	17
10	3	19	13

What we would like to do is have this image input to a circuit that has other neurons and their associated synapses and keep track of what happens. To do this we need to use a *vector* which lists all the values. To cue that it is a list, the convention is to use a boldface symbol, i.e.,

$$\mathbf{x} = (3, 7, 15, 18, 6, 1, 20, 22, 2, 5, 8, 17, 10, 3, 19, 13)$$

Now if that image were represented by labeled lines, then each such line, arriving at a neuron, would make contact with a synapse. Supposing that there are also 16 of them, they can also be represented as a vector

$$\mathbf{w} = (w_1, w_2, w_3, \dots, w_{16})$$

Where the w s represent the strength of the synapse. In a simple model, the effect of such a contact is to multiply the image component x_k with the associated synapse strength w_k . Then the total effect of all the contacts is just the sum of each such interaction. Call this the response of the neuron, r . This can be written as:

$$r = \sum_{k=1}^{16} w_k x_k = w_1 x_1 + w_2 x_2 + \dots + w_{16} x_{16}$$

Conventionality we write this very tidily as

$$r = \mathbf{w} \cdot \mathbf{x}$$

This the *dot product* of two vectors.

An important property of a vector is its *length*, denoted $\|\mathbf{x}\|$ and computed as

$$\|\mathbf{x}\| = \sqrt{x_1^2 + x_2^2 + x_3^2 + \dots + x_{16}^2}$$

We can multiply our vector by a number α which changes its length, i.e.

$$\alpha \mathbf{x} = (\alpha x_1 + \alpha x_2 + \dots \alpha x_{16})$$

Now the final concept that we will need is that of a *projection*. You can think of the vector \mathbf{w} as a coordinate axis, like the familiar x-y-z coordinates in three-dimensional space. In three dimensions the components (x_1, y_1, z_1) are actually the projections of the three dimensional vector onto such axes. This works out so simply in the 3D case because by convention the axes are all perpendicular to each other. However in the general case it is possible to use an arbitrary number of vectors that need not be perpendicular. Nonetheless the computation of the

projections remains the same: construct the perpendicular from \mathbf{x} to \mathbf{y} . Then the projection is given by

$$||\mathbf{x}|| \cos \theta$$

where θ is the angle between the two vectors. Anticipating what is coming next, simplify the notation by letting $\alpha = ||\mathbf{x}|| \cos \theta$.

The gist of the algorithm for learning receptive fields is extraordinarily simple. When an input arrives, the neuron whose synapses are most similar to it are made even more similar. All that remains is to say how this is done mathematically. Suppose the synapses of V1 neurons were initially random values and let us just concern ourselves with a tiny 8×8 image patch, so that $n = 64$, and the images values are given by $\mathbf{x} = (x_1, x_2, \dots, x_{64})$. Since there are now lots of neurons, denote the receptive field of the k th one by \mathbf{w}_k . For each one, compute its response, which we can do as

$$r_k = \mathbf{x} \cdot \mathbf{w}_k$$

Now pick the neuron with the highest response, let's call it k^* , and adjust its receptive field to make it a little more like the image that caused it to win the competition. That is we can change its synapses by a small amount $\Delta \mathbf{w}_k^*$. So the algorithm is given by:

ImageNum = 0	Set the no. of image patches to zero
While ImageNum < Total Images do	Repeat the indented instructions
$\mathbf{x} = \text{GetImageSample}$	Take a patch from an image database
Foreach k compute $r_k = \mathbf{x} \cdot \mathbf{w}_k$	Compute the similarities
Pick the k^* that maximizes r_k	Find the neuron most similar
$\mathbf{w}_k^* = \mathbf{w}_k^* + \alpha(\mathbf{x} - \mathbf{w}_k^*)$	Move the neuron towards the winner
ImageNum = ImageNum + 1	Get ready for another patch

What this algorithm does is create single neurons that approximate as best they can the images that can arise in the natural world. The question is: For small image patches, what is the average patch like? The answer is that it looks like an 'edge,' a light-dark transition. These receptive field patterns were dubbed 'simple cells' by Hubel and Wiesel who first found them.

The algorithm is not finished for a couple of reasons, one relatively unimportant and one important. As for the first, it turns out that in the competition for matching the image some neurons never get picked and so they never develop receptive fields. This can be fixed by defining proximities and allowing the cells near the winner in response to adjust their receptive fields also. The more interesting fix is that by allowing additional neurons to participate in the image patch approximation, we can increase the library of receptive fields to get much closer approximations to the stimulus. As a bonus, this modification produces the kinds of sets seen in Figure 3.9. The trick is to allow the subsequent neurons to approximate the leftover or *residual* after the best neuron

has done its job. In other words, after fitting an image, subtract the fit and have the neurons compete to represent the information in the residual. This can be defined mathematically by starting with $r_0 = 0$, $\mathbf{w} = \mathbf{0}$ and for $k > 0$ computing the following responses sequentially. Let w_k^n denote the winner for competition round n . Then for $n = 0$,

$$r_0 = \mathbf{x} \cdot \mathbf{w}_{k^0}$$

as before. Now let's bring the other neurons that are going to work on the residual into play:

$$r_n = (\mathbf{x} - \mathbf{y}_n) \cdot \mathbf{w}_{k^n}$$

where \mathbf{y}_n is the best approximation so far, represented by

$$\mathbf{y}_n = \sum_{i=0}^{n-1} r_i \mathbf{w}_{k^i}$$

How many residuals should one use? Since the patch has 64 numbers, to be sure of approximating it perfectly it would take 64 neurons, but it turns out that almost all natural images can be approximated with much fewer, say 8–12 neurons. Furthermore the simulations that use fewer patches are the ones that produce the biological-looking receptive fields. Why is this so?

While we do not know the answer to the question of why codes that use small numbers of neurons are biological, a very intriguing suggestion is that the brain is trading off the cost of coming up with the neurons against the accuracy of the approximation. Remember that the response is going to be signaled somehow with spikes and signaling with spikes is very expensive; half the metabolic cost of the neuron. So if there are a 'good enough' approximation that are much cheaper than the very accurate ones the pressure to choose them would be very great indeed. Furthermore- hold onto your hat - a grand way of thinking of this strategy is that the brain is engineered to come up with a theory of itself. It chooses theories that work in that they account for what happens in the world, but that an additional feature is that the cost of the theory itself, here in spikes, must be taken into account. The way this is done is by limiting the number of vectors that are used in the approximation.

The thrust of this section was to show that the receptive field of a cell need not be specified in the genes per se, but rather once a basic circuit connectivity is set up, it can learn these values using a coding principle. The algorithm is purely geometrical and since the circuitry in the cortex contains repeated units just like it, we might expect to find it used over and over again as a general strategy. But once the receptive fields are set up, they can be used in general decision making and in fact are, and that is the topic of the next section.

3.8 Neurons and Behavior

The simplest behavior we have is the reflex circuit that links sensation to action. Tapping the knee in the vicinity of the patella will cause the knee to jerk forward

suddenly. This response does not need the forebrain at all. The sensors that detect the mild tap are connected almost directly to neurons in the spinal cord that enervate the muscles that do the retracting. Sherrington's discovery of the reflex arc was enormously influential and provided a way of thinking about more complicated circuitry that did involve the forebrain. Sensory neurons analyzed the stimulus and extracted features. These neurons in turn were connected to motor neurons that triggered an appropriate response. This feature neurons in this line of thinking model were anticipated in the experiments on frogs by neurosciences's pioneer Adrian:

"I had arranged electrodes on the optic nerve of a toad in connexion with some experiments on the retina. The room was nearly dark and I was puzzled to hear repeated noises in the loudspeaker attached to the amplifier, noises indicating that a great deal of impulse activity was going on. It was not until I compared the noises with my own movements around the room that I realized I was in the field of vision of the toad's eye and that it was signalling what I was doing.
 . . ."

Adrian's observation famously was made more concrete by Lettvin's experiments on the frog's retina that revealed neurons that responded selectively to dark moving spots. The spots were just the right feature to describe moving insects. These and other experiments have shown that a frog has rather stereotypical behaviors that use features that are directly computed in the retinas. Such behaviors are so regular that they can be effectively modeled. Figure 3.11 shows one such model. The early sensory neurons provide an estimate of the bearing of potential prey in terms of a distance and heading. If the distance is too far, the frog will advance. If the heading is not appropriate for the frog's tongue snapping reflex, it will orient and finally, if both those conditions are met, a snap will occur.

The computation of features at the beginning of the circuitry is a feature of other animals too. A rabbit's retinas contain neurons that compute features that measure three-dimensional rotation. These feed circuits that also include inputs from the rabbit's vestibular system that is measuring head accelerations. The objective of the circuit is to make sure that the rabbit keeps very still under certain circumstances so as to be less detectable by predators.

When we study primates though, we find that the neurons that are representing these kinds of features are computed by more central circuitry in the cortex. Why is this so? One idea is that in the cortex the features can be used in many different ways by combining them flexibly in programs, whereas when they are found in the retinas as in other species, they cannot be flexibly combined. But at any rate they can still be used in the role of feature detectors in decision making. A spectacular demonstration of this was done by Newsome, who trained monkeys to make decisions based on moving dot stimuli. Neurons that do the frog's job of detecting motion are found several synapses into the cortex in a cortical region called medial temporal cortex or simply MT. Monkeys were trained to signal with eye movements whether the dots in a pattern

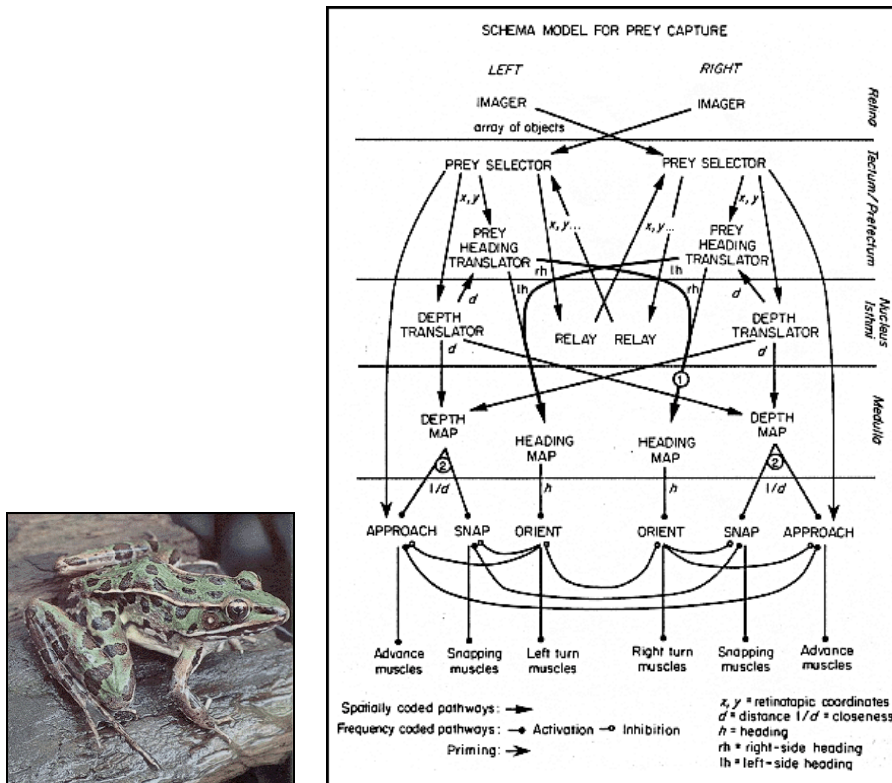


Figure 3.11: A frog has a regimented prey-finding strategy that is modeled in this circuit diagram. The centerpiece is a reflex: if neurons in the visual field signal a moving object at the right size and depth, the frog snaps its tongue out to get it. If the prey is too far away the frog will approach it. If the prey is out of reach of the tongue the frog will orient towards it

of moving dots were moving up or down. Newsome laced the stimulus with randomly moving dots and varied the percentage of them to make the problem successively more difficult. His data showing the different spike rates are shown in Figure 3.12.

Because spike rates are low, experimenters use the *histograms* to sum up the results of many different trials and thus emphasize small differences. Thus the four panels on the right of the figure show individual trials with black dots indicating spikes and then the sum of these trials immediately below. Strong motion differences elicit very pronounced differences in spike rates, but even very weak differences still show up as revealed in the two upper sets of traces.

The monkeys made successively more errors with the more difficult patterns. Amazingly, when he recorded the responses of neurons in a motion sensitive part of the cortex, the individual neurons had firing rates that increased in a way

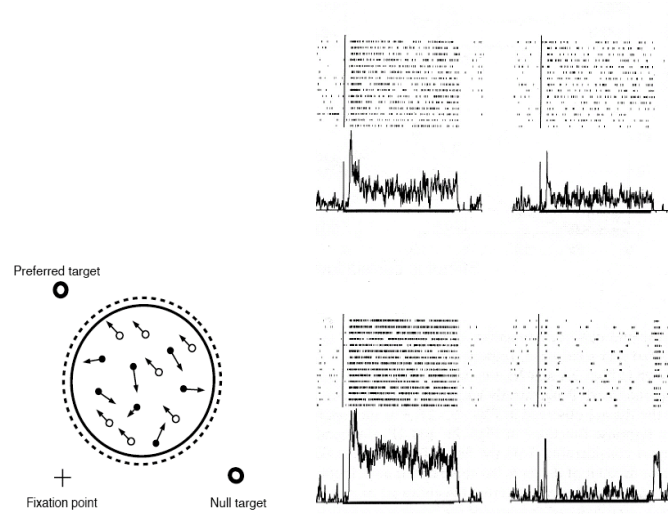


Figure 3.12: (*left*) A monkey sees a pattern of moving dots, some of which are moving to the upper right. If the monkey thinks the motion is in the direction preferred by the cell he must make an eye movement to the spot on the upper left. Otherwise the eye movement should be to the lower right. (*right*) The responses of a neuron in a cortical area sensitive to motion vary as the number of coherently moving dots. Upper panel: response to 25% coherence in preferred and opposite directions. Lower: response to 99% coherence.

that paralleled the success of the monkey on patterns in the task. In a coup de grace of experimental artistry, stimulating a "down motion" neuron when the pattern was up motion so that it sent out more spikes made the monkey reverse its normal decision.

While on the surface it appears that the motion neurons in area MT are performing the same kind of function as those in the frog's retina there is a very important difference in that the cortical neurons were trained to do this job. The circuitry for making decisions about the particular representation of motion on the stimulus screen was no pre-wired but instead was formed, presumably by changing synaptic strengths, into one that would do the job. Given the vast array of different neurons in the cortex, this experiment hints at the huge flexibility the cortex has in creating different states that can be used by programs. How that might happen is the subject of the next chapter.

3.9 Summary

Key ideas:

1. Cell ionic transport is increasingly well understood

2. New data show how synapses are constructed
3. Synapses are the key to programs
4. Only a finite number of neurons puts premium on encoding
5. Feedforward view has codes as patterns
6. Codes may reflect coding cost
7. Exact signaling strategy is still under study
8. Rate code is a model that works in some cases

Bibliography

- [1] J. M. Allman. *Evolving Brains*. W. H. Freeman, 1998.
- [2] A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology*, 1952.
- [3] Peter Lennie. The cost of cortical computation. *Current Biology*, 2003.
- [4] John G. Nichols, A. Robert Martin, and Bruce G. Wallace. *From Neuron to Brain: A Cellular and Molecular Approach to the Function of the Nervous System*. Sinauer, 1992.