2 Strategies for Finding Neural Codes

Sheila Nirenberg

Summary

A critical problem in systems neuroscience is determining what the neural code is. Many codes have been proposed—coarse codes, fine codes, temporal correlation codes, and synchronous firing codes, among others. The number of candidates has grown as more and more studies have shown that different aspects of spike trains can carry information (reviewed in Averbeck and Lee, 2004; Oram et al., 2002; Victor, 1999; Borst and Theunissen, 1999; Johnson and Ray, 2004; Theunissen and Miller, 1995; Nirenberg and Latham, 2003; Shadlen and Newsome, 1994; MacLeod, Backer, and Laurent, 1998; Bialek et al., 1991; Nirenberg et al., 2001; Parker and Newsome, 1998; Romo and Salinas, 2001; Dhingra et al., 2003; Gawne, Richmond, and Optican, 1991).

Here we present a strategy to reduce the space of possibilities. We describe a framework for determining which codes are viable and which are not, that is, which can and cannot account for behavior. Our approach is to obtain an upper bound on the performance of each code and compare it to the performance of the animal. The upper bound is obtained by measuring code performance using the same number and distribution of cells that the animal uses, the same amount of data the animal uses, and a decoding strategy that is as good as or better than the one that the animal uses. If the upper-bound performance falls short of the animal's performance, the code can be eliminated. We demonstrate the application of this approach to a model system, the mouse retina.

Introduction

Finding the neural code has been a long-standing problem. While we have known for decades that neural signals come in the form of trains of action potentials, we still do not know what the code is, that is, we do not know what the relevant quantity

in the spike train is. Is it the number of spikes produced over some behaviorally relevant time period (e.g., the length of a saccade, a "coarse" code), or is it the individual spike or some pattern of spikes (a "fine" code)?

This question has been the subject of much discussion and debate, since the answer affects essentially all work in systems neuroscience. For experimental work, it tells us the resolution we should use when analyzing our data, and for theoretical work, it tells us the quantity (the spike train feature) we should use when building models for neural computations. It also provides critical constraints on the structural design of such models.

Just to expand on this a little bit: Even if one thinks about it in the simplest terms, it becomes clear that finding the code (or codes) is a key step toward understanding how neural systems are designed. This is because different codes have different requirements for neural machinery, i.e., for implementation, for readout, and so on, and these differences lead to different models for how the system is set up. For example, take a simple case: a coarse code versus a fine code. These two codes have different strengths and weakness, and these differences have obvious implications for network structure. For example, one of the strengths of coarse codes is that the details of the spike train do not matter. This means that downstream neurons do not have to keep track of the spike arrival times; they just have to count spikes. The weakness, though, is that coarse codes cannot, at an individual cell level, carry much information. So, to get enough information to represent the outside world, which has, of course, many stimuli in it, one would almost certainly have to postulate a pooling mechanism, some neural machinery that would allow downstream neurons to pool signals from multiple cells or across time or both. Without this, coarse coding would not be viable. In contrast, with fine coding, much more information can be carried. The number of stimuli that can be represented is much larger, exponentially so, but fine codes impose their own constraints: downstream neurons *do* have to keep track of spike arrival times.

Thus, even at this simple level, the importance of finding the code(s) is clear. For more complex examples about what specific codes imply about network structure—or the reverse, what specific network structures imply about the code—see analyses of synfire chains (Abeles et al., 1993) and balanced networks (van Vreeswijk and Somplinsky, 1996), reviewed in van Vreeswijk (2004).

A Strategy for Finding Neural Codes

It might seem, at first, that finding the code—at least for a given brain area—is a straightforward problem, one that could be addressed as follows: Give an animal a task to perform and measure its performance; then take the spike trains the animal

uses to perform the task and decode them several times, each time making a different assumption about what the relevant quantity is. For example, first assume it is the number of spikes in a relatively long time period, such as the length of a stimulus presentation or a saccade, then assume it is the number of spikes in a shorter time period, etc. Then, with each assumption, measure how well the task was performed based on the spike trains and compare it to the performance of the animal.

While this approach is straightforward in principle, it is not so straightforward in practice. This is because several conditions have to be met in order to draw clean conclusions. First, the number and distribution of cells used for the decoding has to be the same as the number and distribution of cells the animal uses. To see why this matters, take coarse coding as an example. As mentioned above, with coarse coding, individual cells by themselves cannot carry much information, but, together, as a population, they can and this population information could be sufficient. So unless one records spike trains from *all* the cells the animal uses to solve a task, one cannot reject the coarse coding hypothesis and assert that a finer code—one that carries more information per cell—is necessary. (From an experimental viewpoint, recording from *all* the cells remains challenging; see, however, the approach herein with respect to retinal ganglion cells.) Second, the time period over which the spike trains are evaluated has to be the same as the time period the animal uses. For example, when an animal evaluates a stimulus, it typically looks at it several times. Unless data are collected from multiple looks as well as multiple cells, again, codes cannot be eliminated. Finally, the last condition is that the decoding algorithm used to test codes has to be at least as good as the one the animal uses. Since no one knows the algorithm the animal uses, the only option is to use optimal, that is, Bayesian, decoding—a strategy that extracts as much information from the spike trains as can be extracted (Gelman et al., 1995).

If these conditions are met, one gets an upper bound on the performance of a code. The code is, essentially, being given its best chance—it is being tested using the same number and distribution of cells the animal uses, the same amount of data as the animal uses, and using a decoding algorithm that is as good as or better than the animal's. If one has an upper bound on the performance of a code, and that upper bound falls short of the animal's performance, then one can rule that code out.

Application of the Strategy to a Model System

In this section, we describe a set of experiments where it was possible to meet these conditions (Nirenberg, 2006; Jacobs et al., 2009). One of the few places this can be done is the retina. The reasoning is as follows: First, the retinal output cells, the

ganglion cells, form one of the few bottlenecks in the nervous system—they are the sole source of visual information to the brain. This means that all the cells needed for the recording are confined to one small, well-defined location. Second, the information flow from the retina to the brain is feedforward. Because there is no feedback, it is possible to remove the retina from the animal and record from it in vitro. This is a major advantage, because it allows one to use multi-electrode arrays, which make it possible to obtain a large dataset, that is, one large enough to match the number and distribution of cells the animal uses (Balkema and Pinto, 1982; Carcieri, Jacobs, and Nirenberg, 2003; Stone and Pinto, 1993). Third, the number of cells one needs to record from can be controlled by regulating the size of the stimulus: a stimulus subtends a certain number of degrees of visual angle, which corresponds to a known area on the retina, and, therefore, a known number of ganglion cells (Remtulla and Hallett, 1985). Finally, the time period over which data need to be collected can be controlled by regulating the duration of the stimulus and the number of times it is presented.

With these conditions met, we set out to test the performance of a set of widely proposed codes using a combined in vivo and in vitro approach. We started *in vivo*. We gave animals a task to perform, as shown in figure 2.1. The animal was the mouse, and the task was a two-alternative, forced-choice visual discrimination task (Prusky et al., 2000) that proceeded as follows: On each trial of the task, two stimuli were flashed on. One was a uniform gray field, the other, a sine wave grating, where the spatial frequency of the grating was chosen at random from a set of uniformly spaced frequencies. The animal's job was to determine which of the two stimuli was the grating. A schematic of the task is shown in figure 2.1a, and the behavioral performance for a set of eight animals is shown in figure 2.1b. As shown in the figure, when the spatial frequency of the grating was low, the animal performed well (i.e., the fraction correct was at or near 100 percent). As the spatial frequency increased, the grating and gray became increasingly difficult to distinguish, and performance dropped, eventually to chance (50 percent correct).

To measure code performance, we presented the same task to the retina in vitro, matching all stimulus conditions (stimulus size on the retina, stimulus presentation, intensities, spatial frequencies, phases, contrasts, and position, see Jacobs et al., 2009). We then decoded the ganglion cell spike trains several times, each time making a different assumption about what the code is, that is, what the relevant quantity in the spike train is. We started with a simple coarse code, a spike count code. Here the assumption was that the relevant quantity is just the number of spikes in a stimulus presentation. We then advanced to a finer code, a spike timing code, also referred to as an "instantaneous rate" code. Here the assumption was twofold—that the relevant quantity is, essentially, the individual spike, and that the occurrence of a spike does not depend on the occurrence of other spikes. Last, we advanced to a

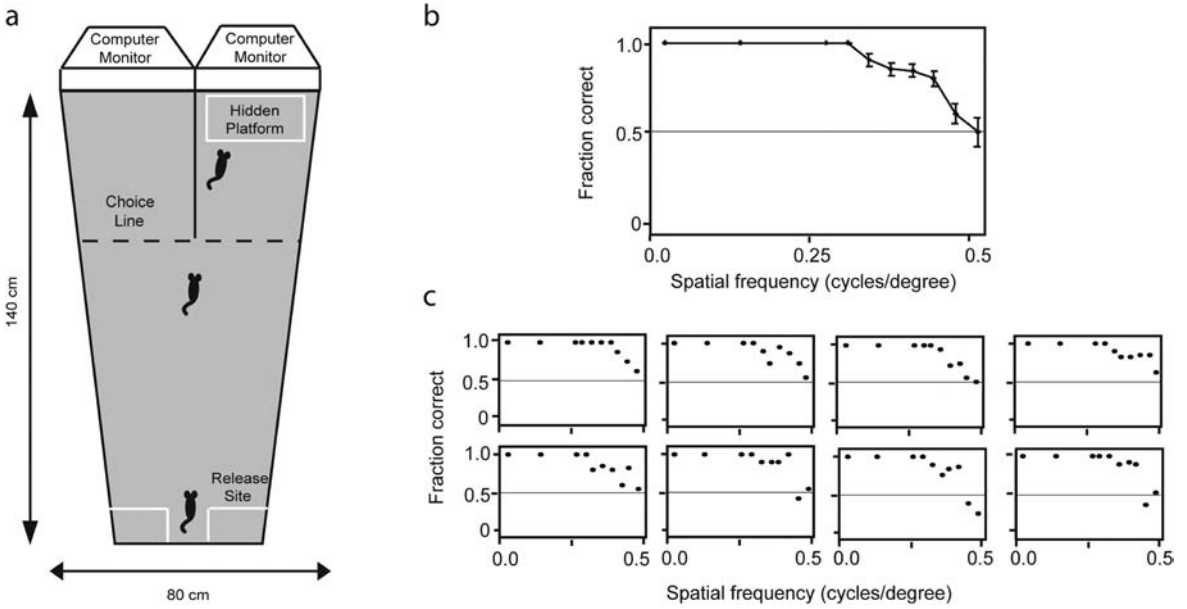

Figure 2.1
Performance of the animal on a two-alternative, forced-choice visual discrimination task. (a) Schematic of the task. On each trial the animal is dropped into a pool, where the only escape is a hidden platform. The animal can determine the location of the platform by viewing the stimuli on the computer monitors at the front of the pool. One monitor will show a grating, the other a uniform gray field. The platform is always placed on the side with the grating. If the animal can determine which of the two stimuli is the grating, it can find the platform and escape. For these experiments, each stimulus, which subtended 10×15 degrees of visual angle at the choice line (see figure), was flashed on eight times, and each flash lasted for 300 ms, shorter than the time between saccades. A barrier was set up so that the animal could see only one stimulus at a time. (b) Mean performance for eight animals, plotted as the fraction of times the animal chose the grating as a function of spatial frequency. Error bars were computed using binomial statistics: the standard deviation was $[p(1 - p)/n]^{1/2}$, where p is the probability of choosing the grating and n is the number of trials (n ranged from 10 to 115 trials/spatial frequency). (c) Performance of each animal shown separately. Figure adapted from Jacobs et al. (2009).

temporal correlation code. For this, the underlying assumption was also that the relevant quantity is the individual spike, but this time the occurrence of a spike *was* assumed to depend on the occurrence of other spikes—in this case, we made it depend specifically on the time of the previous spike. (See the appendix, where a mathematical description of each code is provided.) In sum, then, we decoded spike trains using three widely proposed codes—a spike count code, a spike timing code, and a temporal correlation code—and measured each one's performance against the performance of the animal.

Figure 2.2 shows the results. The first result was that the spike count code performed much worse than the animal ($p << 0.0001$, psignifit test) (Wichmann and

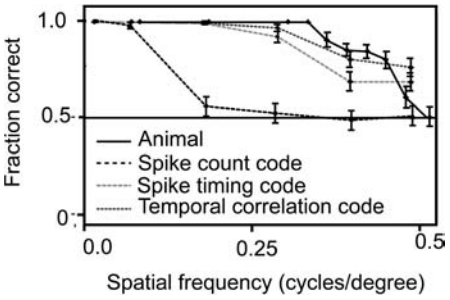

Figure 2.2
The spike count code and spike timing code performed worse than the animal, while the temporal correlation code did not (spike count code: $p << 0.0001$; spike timing code: $p < 0.02$; temporal correlation code: $p > 0.3$). As in figure 2.1, error bars were computed using binomial statistics: the standard deviation was $[p(1 - p)/n]^{1/2}$, where p is the probability of choosing the grating and n is the number of trials (for the codes, $n = 98$ trials/spatial frequency; for the animals, n ranged from 10 to 115 trials/spatial frequency). Figure adapted from Jacobs et al. (2009).

Hill 2001a, 2001b). While the animal performed the task well (>70 percent correct up to 0.4 cpd), the spike count code was only able to maintain this level up to about 0.1 cpd. This argues that the animal cannot be using this code. The claim is strong because, as mentioned earlier, the code was given its best chance, that is, it was assessed using the same number and distribution of cells the animal uses, taking data for as long as the animal does, and using a decoding strategy that can extract as much information from the code as can be extracted. This means that for the animal to be performing as well as it does, it has to be using a more information-rich code. This latter assertion follows because there is no other way for the animal's brain to be getting information: (1) as mentioned before, the retinal output cells are the only source of visual input to the brain, and (2), the brain cannot create information *de novo*. As stated by the data processing inequality, a well-known theorem in signal processing (Cover and Thomas, 1991) information cannot be generated by postprocessing. A system can manipulate the information it receives, perform computations on it, and so forth, but it cannot create new information.

The second result was that the spike timing code also performed worse than the animal ($p < 0.02$). This argues that this code is not viable either. It does, though, perform considerably better than the spike count code. If one views it from a practical level, that is, from the point of view of building a prosthetic, one could say that a prosthetic constructed using a spike count code would fall seriously short, whereas one built with a spike timing code would put the animal in the ballpark of normal acuity.

Finally, the last result was that the temporal correlation code *did* perform the task as well as the animal ($p > 0.3$, psignifit test). While this result does not prove that

this is the code the animal uses, it does show that it carries sufficient information and constitutes a viable candidate code. To see different code performances at the raw data level, see figure 2.3 (plate 2).

These results made strong claims, that two of the most widely proposed codes—the standard spike count and spike timing codes—are not viable. How confident could we be of these claims? The answer depended on how well they stood up to potential errors we might have made in the estimates of the critical parameters: specifically, the estimates of cell number, cell distribution, priors on the stimulus, and shapes of the response distributions. The first source of potential error was the estimate of the number of ganglion cells the animal uses. The stimulus covers 0.144 mm^2 of retina. Two recent electron microscopic estimates of ganglion cell number (Jeon, Strettoi, and Masland, 1998; Williams et al., 1996) indicated a range of 300 to 360 cells for this area. We measured the performance of each code using both numbers, and there was essentially no difference (figure 2.4a). The figure shows performance as a function of cell number, and, as indicated in the figure, performance growth slowed down at numbers much lower than these.

The second source of potential error was in the estimation of the ganglion cell distribution. Physiological reports suggest that the distribution is skewed toward ON-type ganglion cells (Balkema and Pinto, 1982; Carcieri, Jacobs, and Nirenberg, 2003; Stone and Pinto, 1993; Sagdullaev and McCall, 2005), but anatomical studies suggest a more even representation, specifically, a more even representation of ON and OFF-type cells (Doi et al., 1995). The latter is based on the relative density of the projections into the ON and OFF layers in the innerplexiform (processing) layer, which reflects the relative proportions of cells with ON and OFF responses (Doi, Uji, and Yamamura, 1995). We measured the performance of each code, building the distribution of cell types both ways (figure 2.4b). While the choice of distribution shifted the performance slightly, the conclusions remained the same: the spike count and spike timing codes still performed worse than the animal (spike count, $p << 0.0001$; spike timing code, $p < 0.02$); the temporal correlation code did not perform worse than the animal ($p > 0.3$).

The third issue is the estimation of stimulus priors. We measured performance using both uniform priors ($p(\text{gray}) = 1/2$; $p(k)$ is a constant, where k is spatial frequency) and natural priors ($p(\text{gray}) = 1/2$; $p(k) \propto 1/k^2$), the latter following from Ruderman and Bialek (1994). As shown in figure 2.4c, column 3, there was essentially no effect: the spike count and timing codes still performed worse than the animal (spike count, $p << 0.0001$, spike timing, $p < 0.02$); temporal correlation code did not perform worse than the animal ($p > 0.3$).

Finally, the last issue concerns the estimation of the response distributions. Because we are using a Bayesian decoding approach, the response distribution for each stimulus has to be estimated, and its quality depends on the number of

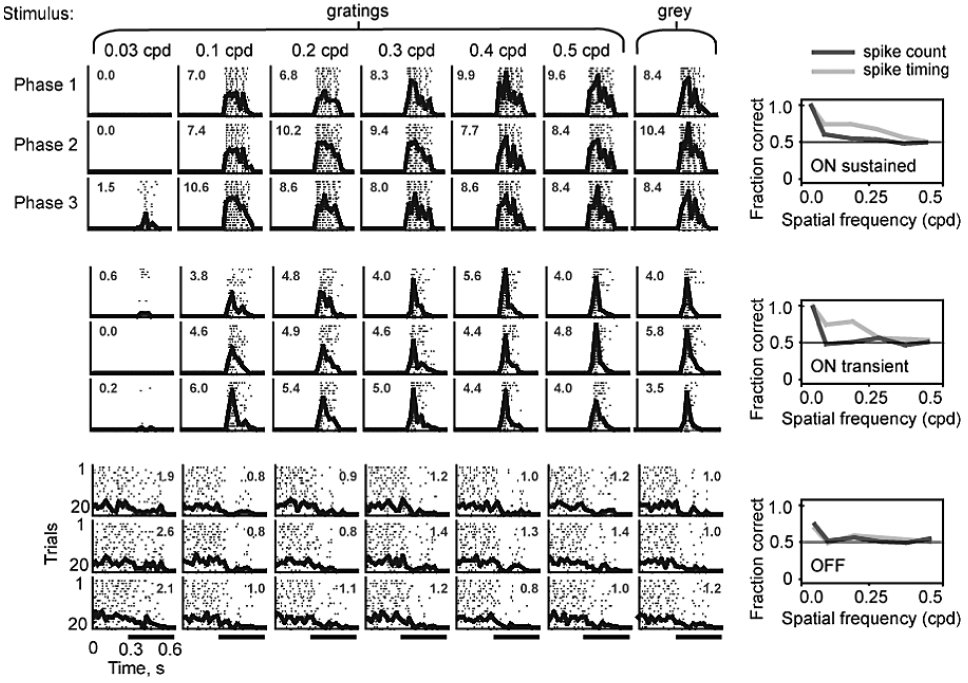

Figure 2.3 (plate 2)
Rasters for three cells; the purpose is to show that the differences in the performances of the codes can be seen at the level of the raw spike trains. Each set of 21 rasters corresponds to 1 cell. On each raster are a number and a red trace. The number is the mean number of spikes per trial for the raster and gives a measure of spike count, and the red trace is the peristimulus time histogram, and gives a measure of spike pattern. To get a feel for how a given performance curve arises (see blue and green performance curves in the plot at the right), one can compare the mean spike counts for the gray stimulus with the mean spike counts for each of the spatial frequencies, and then compare the mean spike patterns for the gray stimulus with the mean spike patterns for each of the spatial frequencies. For example, for the first cell, the mean spike counts for the three rasters at the lowest spatial frequency, 0, 0, and 1.5, are very different from those for the gray stimulus, 8, 10, and 8. As the spatial frequency gets higher, though, the mean spike counts for the gratings become less and less different from those for the gray. In contrast, the spike patterns for the gratings remain visibly different out to much higher spatial frequencies. Thus, where spike count starts to fail as a useful parameter for distinguishing among stimuli, spike pattern continues to perform. *Methods*: To remind the reader, the stimuli are flashed. Between flashes the screen is black; thus, each stimulus produces a step increase in mean illumination. Note that the phases are a small fixed distance apart (50 microns) to mimic normal displacements due to eye movements (rather than equally spaced phases).

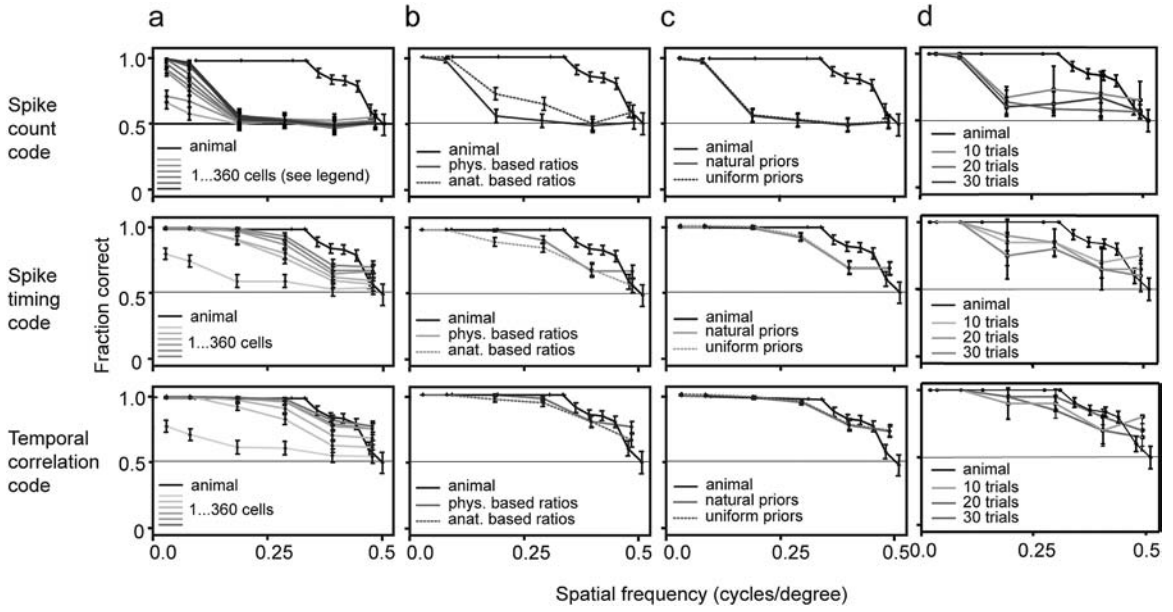

Figure 2.4
The failure of the spike count code and the spike timing code was robust to potential errors in the estimation of the critical parameters: cell number, cell distribution, priors on the stimulus, and shapes of the response distributions. (a) Performance of spike count and spike timing codes remained worse than the animal with cell numbers up to 360 (the upper bound estimate on cell number, see text), whereas performance of the temporal correlation code reached that of the animal (was not significantly different ($p > 0.3$). For spike count, the traces indicate 1, 2, 4, 8, 16, 32, 64, 128, 256, 300, and 360 cells. For the spike timing and temporal correlation codes, the traces indicate 1, 64, 128, 256, 300, and 360 cells. Note that for all three codes, performance became very slow growing at numbers well below 360 suggesting a saturation in performance (for this task) with relatively small numbers of cells. Error bars were computed as in figure 2.2. (b) Performance for all codes shifted slightly when the distribution of cell classes was drawn from anatomical versus physiological estimations, but the conclusions remained the same: the spike count and spike timing codes still performed worse than the animal ($p \ll 0.0001$, $p < 0.02$, respectively); the temporal correlation code did not perform worse than the animal ($p > 0.3$). Error bars were computed as in figure 2.2. (c) Performance of the codes remained essentially the same whether uniform or natural priors were used; again, all conclusions remained the same. Error bars were computed as in figure 2.2. (d) No significant trend in performance occurred when the number of trials was systematically varied. The numbers 10, 20 and 30 on the figure indicate the number of trials used to build the response distribution for each phase of each spatial frequency; this gives a total of 30, 40, and 60 trials respectively for each spatial frequency, since each spatial frequency was represented by three phases. The error bars indicate the standard deviation for three cross-validations. Figure adapted from Jacobs et al. (2009).

responses. Too few repetitions could lead to a misestimate of code performance. To address this, we ran the analysis such that the response distributions were built with different numbers of stimulus repeats (figure 2.4d). The results show that for the spike count and spike timing codes, there was no significant trend as the number of repeats was increased, that is, the performance of these two codes did not significantly change, and both remained below the performance curve of the animal. For the temporal code, there was also no significant trend as the number of repeats increased, but, here, nearly all points in the performance curves reached the animals' behavioral performance curve. This supports the claim that this code cannot be ruled out and stands as a viable candidate code. We also performed this analysis with multiple cross-validations; this is represented by the error bars in panel *d*. This further demonstrates the robustness of the results: even when the variance that occurs with different cross validations in taken into account, the differences in the performances of the codes is clear: the spike count code performs considerably worse than the animal, the spike timing code performs slightly worse, and the temporal correlation code reaches the animals' performance.

In sum, we described here a strategy for testing the viability of neural codes. The approach was to obtain an upper bound on the performance of each code and compare it to the performance of the animal. The upper bound was obtained by measuring code performance using the same number and distribution of cells the animal uses, the same amount of data the animal uses, and a decoding strategy that is as good as or better than the one the animal uses. If the upper bound performance fell short of the animal's performance, the code was eliminated, as this indicated very strongly that the animal cannot be using it.

We tested three widely proposed codes, and our results showed that two of them, the spike count and spike timing codes, did, in fact, fall short. Interestingly, the performance of the spike count code fell substantially short, as shown in figures 2.2 and 2.4. This result also held when we counted spikes in windows smaller than the length of the stimulus presentation. This shows that the failure of this code was not being exaggerated by counting spikes in the full 300 ms window. Even when spikes were counted only in 100 ms and 50 windows, the spike count code performed substantially worse than the animal (figure 2.5). The second result was that the spike timing code also fell short, although the failure of this code was much less dramatic (see figures 2.2 and 2.4). Finally, the last result was that the temporal correlation code did perform as well as the animal. While this does not demonstrate that this is the code the animal uses, it does show that it carries sufficient information and constitutes a viable candidate code. We emphasize, though, that we only tested a small number of codes; other spike pattern permutations (e.g., a coarse code with temporal correlation, a code with multicell correlations [e.g., see chapters 3 and 21] remain candidates for testing).

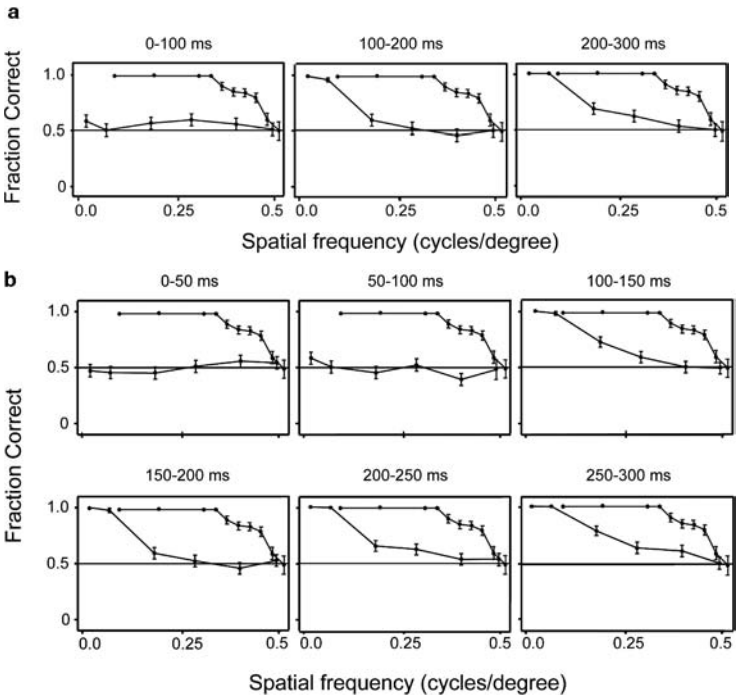

Figure 2.5
Decoding with the spike count code using responses shorter than the complete stimulus period. In figures 2.2 and 2.3, we evaluated the spike count code for the whole stimulus presentation, which was 300 ms long. Since not all of the 300 ms response may contain informative spikes, this raises the possibility that we might be "diluting" the performance of the spike count code, that is, we might be underestimating its performance by including response periods that are essentially only adding noise. Here we measured performance using epochs within the 300 ms response. The results show that with some epochs the performance showed an increase relative to the performance with the complete 300 ms response, but the increase was very small: In all cases the performance fell substantially short of the performance of the animal ($p << -0.001$ for all tests), supporting the notion that the failure of the spike count code is a robust result. Note that these performances were expected to fall short of animal performance, because, barring data limitation issues, the spike count code must do worse than the spike timing code, and the spike timing falls short of the animal (as shown in figures 2.2 and 2.3). The data presented here are just to show the magnitude of the failure with different epochs. (a) Performance when spikes were counted only for first 100 ms, then only for the second 100 ms, etc. (b) Performance when spikes were counted only for first 50 ms, then only for the second 50 ms, etc. Upper trace: performance of the animal; lower trace: performance when spikes were counted only for the period indicated at the top of the panel.

These findings about the neural codes have significant implications for how downstream neurons must perform their computations—they argue that simple coarse coding algorithms built around spike counting, pooling, and so on, are not realistic, at least at the retina/brain interface, and new models—those that take into account additional features of the spike train—need to take their place, as these additional features, such as temporal correlations, carry essential meaning.

The findings also raise the intriguing issue of generalization. The problem of finding the neural code has often been likened to the problem of finding the genetic code, but while there is one genetic code (the relevant quantity or "unit of information" is always 3 nucleotides = 1 codon), it is not clear that there will be only one neural code. The results we presented here apply to the transfer of information from a periphery to the brain, a transfer that may require a particularly information-rich code. There is the intriguing possibility that the brain switches coding strategies when faced with problems with different constraints (e.g., motor control).

Other Indexes for Testing the Viability of Neural Codes

In the preceding experiments, we used the best performance of a Bayesian decoder as an index to test the viability of neural codes. Another index that can be used for this purpose is Shannon's mutual information. These two indexes represent the extremes of a continuum of indexes that provide rigorous tests of neural codes (Victor and Nirenberg, 2008). A recent study discusses the merits of the different indexes, that is, under what conditions to use them, as they differ in numerous ways, such as their sensitivity to the loss of information that occurs when neural activity is converted into a behavioral response (response discretization), their sensitivity to the decision criterion used in performing a task, as well as in their bias and variance characteristics.

The main findings can be summarized as follows: Shannon-like indexes fall short, that is, they fail to eliminate codes that are nonviable, under conditions where the firing patterns that represent the different stimuli differ substantially in certainty, because these differences get suppressed (i.e., information gets lost) when the code is converted to a behavior. Bayes-like indexes provide stronger tests than Shannon-like indexes under these conditions, but the advantage can depend on knowing what the decision rule for the task is; for quantitative discussion, see Victor and Nirenberg, 2008). With respect to statistical properties, Bayes-like indexes tend to have greater bias and variance than Shannon-like indexes for responses close to threshold, but smaller bias and variance away from threshold.

Appendix

Decoding Ganglion Cell Spike Trains

Briefly, we estimate the probability that a particular stimulus, s, was presented, given that a particular set of ganglion cell responses, r, occurred. We denote this probability $p(s|r)$. We find it by presenting each stimulus repeatedly, recording the resulting ganglion cell responses, and estimating the conditional response distribution, $p(r|s)$. We then use Bayes theorem, $p(s|r) = p(r|s)p(s)/p(r)$, to determine $p(s|r)$ from $p(r|s)$.

We then use $p(s|r)$ to perform the task. In the task, there are two stimuli, a grating and a gray screen. Each produces a set of responses. The question we ask on each trial of the task is: which of the two sets of responses corresponds to the grating? Letting r_1 and r_2 be the responses to the two stimuli, s_1 and s_2 respectively, we answer this by comparing $p(s_1 = \text{grating}|r_1)$ to $p(s_2 = \text{grating}|r_2)$. If the first quantity is larger, we say that r_1 corresponds to the grating; otherwise, we say that r_2 does. (We use half the responses, chosen at random, to generate $p(r|s)$, and the other half to perform the task, i.e., standard cross-validation.)

This approach gives us natural way to test different codes, as different codes correspond to different treatments of r. To test the spike count code, we treat r as spike count; to test the spike timing code, we treat r as a set of spike arrival times and assume that the occurrence of a spike is independent of the occurrences of other spikes; and to test the temporal correlation code, we also treat r as a set of spike arrival times, but, this time, assume that the occurrence of a spike is *not* independent of the occurrences of other spikes; the specific dependence we assumed was a dependence on the time of the previous spike on the same spike train. (Note that the same length of response is used; the difference is in the treatment of the responses.)

The following shows how the response distribution, $p(r|s)$, for each code was constructed: For the spike count code, $r = n =$ {the number of spikes in a stimulus presentation (which was 300 ms long, shorter than the time between saccades). Since each trial of the task involves multiple cells and multiple looks (i.e., multiple stimulus presentations), we use n_{il} to denote spike count from cell i on look l, and write

$$p_{\text{spike count code}}(r|\text{s}) = \prod_{il} p(n_{il}|\text{s}) \tag{2.1}$$

This assumes independence among cells and also across looks.

For the spike timing code, the probability of a response given a stimulus follows that of an inhomogeneous Poisson process. Here r is a list of spike times at resolution dt, denoted t_{ijl} where t_{ijl} is the jth spike on the lth look of cell i, this probability is given by

$$p_{\text{spike timing code}}(r \mid s) = \prod_{l}\left[\prod_{ij} \nu_i(t_{ijl} \mid s)dt\right]\exp\left[-\sum_i \int_0^T dt\nu_i(t|s)\right] \tag{2.2}$$

where $\nu i(t|s)$ is the firing rate of cell i at time t, given that stimulus s was presented, and the upper and lower limits (0 and T) correspond to the start and end of each trial. Equation (2.2) was obtained in the standard way, as follows: Time is discretized into bins of size dt with dt small enough to ensure that that there is at most one spike per bin. One then proceeds though the bins and writes down the probability of observing a spike or "no-spike" in each bin. If there is a spike, the probability is $\nu i(t|s)dt$; if there is no spike (i.e., the bin is empty), the probability is $(1 - \nu i(t|s)dt)$. Multiplying the terms together for all bins and taking the small dt limit gives us equation (2.2). (In the small dt limit, the product of all the terms with $(1 - \nu i(t|s)dt)$ results in the exponential term in equation (2.2).)

For the temporal correlation code, again r is a list of spike times at resolution dt, denoted t_{ijl}, but this time the firing rate has an additional dependence on the time of the previous spike on the same spike train,

$$p_{\text{temporal correlation code}}(r \mid s) = \prod_{l}\left[\prod_{ij} \nu_i(t_{ijl}, \tau(t_{ijl}) \mid s)\right]\exp\left[-\sum_i \int_0^T dt\nu_i(t, \tau(t) \mid s)\right] \tag{2.3}$$

where $\tau(t)$ is the time interval between t and the spike that preceded t on the same neuron (e.g., $\tau(tijl) = tijl - ti,j\text{-}1,l$).

Note that we find $p(r|s)$ in equations (2.1)–(2.3) by factorizing it, that is, we compute it from the marginals. Thus, when we decode, we find $p(s|r)$ from the factorized $p(r|s)$, as given in equations (2.1)–(2.3). The justification for this comes from studies by several different groups (Nirenberg et al., 2001; Golledge et al., 2003; Petersen, Panzeri, and Diamond, 2001; Averbeck and Lee 2003; Oram et al., 2001; Meytlis et al., 2009) that show that factorizing $p(r|s)$ has little or no effect on the estimation of $p(s|r)$ for all r that occur."

Note also that equations (2.1)–(2.3) do not imply that spatial relations are disrupted. (The subscripts on the equations indicate this.) Within each retina, each neuron retains its spatial position relative to the stimulus and relative to the other neurons, as shown in figure 2.6 (plate 3).

Computing Instantaneous Firing Rates for Equations (2.2) and (2.3)

The procedure for finding $\nu(t|s)$ and $\nu(t, \tau(t)|s)$ from spike train data was as follows: For $\nu(t|s)$, we followed the method of Kass and Ventura (2001). Briefly, we presented the stimulus repeatedly, binned the responses at 1 ms, and on each trial determined whether or not there was a spike in each of the bins. Given this trial-by-trial data, we parameterized $\nu(t|s)$ with cubic splines, and used maximum likelihood to estimate the spline parameters. We used cubic splines because they provide a way to

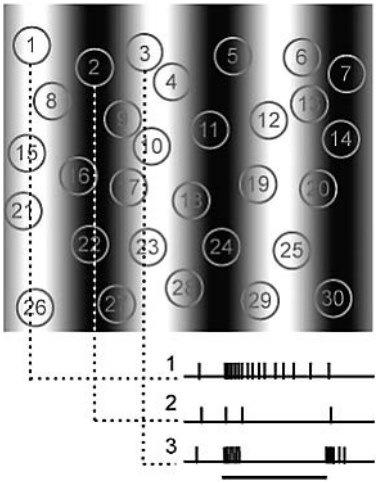

Figure 2.6 (plate 3)
Decoding ganglion cell responses, an illustration. The grating indicates a stimulus, and the circles indicate the receptive field positions of an array of cells relative to it. The traces below show the responses of several of the cells to one presentation of the stimulus. When we decode, we decode the joint response for the population—that is, we decode a vector, whose components are the responses of the cells on that stimulus presentation. When we assume the spike count code, the components are single numbers, spike counts. When we assume the spike timing or temporal correlation codes, the components are sets of spike arrival times. For the formal description of each code, see the appendix.

capture firing rate changes, including sharp transitions, but require a small number of parameters (DiMatteo et al., 2001). In the experiments described here, the ratio of spikes to parameters was large: the mean number of knots (and thus the mean number of parameters) averaged over our dataset was 4.6, with a maximum of 12 and a minimum of 3. Given that we had, on average, 210 spikes/trial, this means we had ~46 spikes per parameter.

To find the firing rate conditioned on previous spikes (as well as stimulus), $r(t, \tau(t)|s)$, we again presented the stimulus repeatedly and determined, on each trial, whether or not there was a spike in each 1 ms bin. We then wrote the firing rate as the product of two terms: $r(t, \tau(t)|s) = r_1(t|s)r_2(\tau(t)|s)$ (Berry and Meister, 1998; Miller and Mark, 1992). Both terms were parameterized, based on the trial-by-trial data, using cubic splines and, again, maximum likelihood was used to estimate the spline parameters. We used four knots for the second term, and the same number (but not placement) of knots for the first. Thus, for the correlation code, we had approximately 24 (210/(4.6 + 4)) spikes per parameter, again, a large spike to parameter ratio. The strong performance of this code compared to behavior suggests that (at least within the current experimental conditions) it is not necessary to look for more complex codes (e.g., those involving higher order correlations among spikes).

Acknowledgments

We thank Jonathan Victor and Chethan Pandarinath for helpful discussion and Cassie Youngstrom for comments on the manuscript. We also thank the Proceedings of the National Academy of Sciences, for several of the figures were adapted from Jacobs et al. (2009).

References

Abeles M, Bergman H, Margalit E, Vaadia E. 1993. Spatiotemporal firing patterns in the frontal cortex of behaving monkeys. *J Neurophysiol* 70: 1629–1638.

Averbeck BB, Lee D. 2003. Neural noise and movement-related codes in the macaque supplementary motor area. *J Neurosci* 23: 7630–7641.

Averbeck BB, Lee D. 2004. Coding and transmission of information by neural ensembles. *Trends Neurosci* 27: 225–230.

Balkema GW, Jr, Pinto LH. 1982. Electrophysiology of retinal ganglion cells in the mouse: a study of a normally pigmented mouse and a congenic hypopigmentation mutant, pearl. *J Neurophysiol* 48: 968–980.

Berry MJ, Meister M. 1998. Refractoriness and neural precision. *J Neurosci* 18(6): 2200–2211.

Bialek W, Rieke F, de Ruyter van Steveninck RR, Warland D. 1991. Reading a neural code. *Science* 252: 1854–1857.

Borst A, Theunissen F. 1999. Information theory and neural coding. *Nat Neurosci* 2: 947–957.

Carcieri SM, Jacobs AL, Nirenberg S. 2003. Classification of retinal ganglion cells: a statistical approach. *J Neurophysiol* 90: 1704–1713.

Cover TM, Thomas JA. 1991. *Elements of information theory*. New York: Wiley.

Dhingra NK, Kao YH, Sterling P, Smith RG. 2003. Contrast threshold of a brisk-transient ganglion cell in vitro. *J Neurophysiol* 89: 2360–2369.

DiMatteo I, Genovese CR, Kass RE. 2001. Bayesian curve-fitting with free-knot splines. *Biometrika* 88: 1055–1071.

Doi M, Uji Y, Yamamura H. 1995. Morphological classification of retinal ganglion cells in mice. *J Comp Neurol* 356: 368–386.

Gawne TJ, Richmond BJ, Optican LM. 1991. Interactive effects among several stimulus parameters on the responses of striate cortical complex cells. *J Neurophysiol* 66(2): 379–389.

Gelman A, Carlin JB, Stern HS, Rubin DB. 1995. *Bayesian data analysis*. London: Chapman and Hall.

Golledge HD, Panzeri S, Zheng F, Pola G, Scannell JW, et al. 2003. Correlations, feature-binding and population coding in primary visual cortex. *Neuroreport* 14: 1045–1050.

Jacobs AL, Fridman G, Douglas RM, Alam NM, Latham PE, Prusky GT, Nirenberg S. 2009. Ruling out and ruling in neural codes. *Proc Natl Acad Sci USA* 106: 5936–5941.

Jeon CJ, Strettoi E, Masland RH. 1998. The major cell populations of the mouse retina. *J Neurosci* 18: 8936–8946.

Johnson DH, Ray W. 2004. Optimal stimulus coding by neural populations using rate codes. *J Comput Neurosci* 16: 129–138.

Kass RE, Ventura V. 2001. A spike-train probability model. *Neural Comput* 13: 1713–1720.

MacLeod K, Backer A, Laurent G. 1998. Who reads temporal information contained across synchronized and oscillatory spike trains? *Nature* 395: 693–698.

Meytlis, M, Bomash, I, Pillow, JW, Nirenberg. S. 2009. Assessing the importance of correlated firing using large populations of neurons. SFN abstract, in press.

Miller MI, Mark KE. 1992. A statistical study of cochlear nerve discharge patterns in response to complex speech stimuli. *J Acoust Soc Am* 92(1): 202–209.

Nirenberg S. 2006. Ruling out neural codes. *J Vis* 6: 889a.

Nirenberg S, Carcieri SM, Jacobs AL, Latham PE. 2001. Retinal ganglion cells act largely as independent encoders. *Nature* 411: 698–701.

Nirenberg S, Latham PE. 2003. Decoding neuronal spike trains: how important are correlations? *Proc Natl Acad Sci USA* 100: 7348–7353.

Oram MW, Hatsopoulos NG, Richmond BJ, Donoghue JP. 2001. Excess synchrony in motor cortical neurons provides redundant direction information with that from coarse temporal measures. *J Neurophysiol* 86: 1700–1716.

Oram MW, Xiao D, Dritschel B, Payne KR. 2002. The temporal resolution of neural codes: does response latency have a unique role? *Philos Trans R Soc Lond B Biol Sci* 357: 987–1001.

Parker AJ, Newsome WT. 1998. Sense and the single neuron: probing the physiology of perception. *Annu Rev Neurosci* 21: 227–277.

Petersen RS, Panzeri S, Diamond ME. 2001. Population coding of stimulus location in rat somatosensory cortex. *Neuron* 32: 503–514.

Prusky GT, West PW, Douglas RM. 2000. Behavioral assessment of visual acuity in mice and rats. *Vision Res* 40: 2201–2209.

Remtulla S, Hallett PE. 1985. A schematic eye for the mouse, and comparisons with the rat. *Vision Res* 25: 21–31.

Romo R, Salinas E. 2001. Touch and go: decision-making mechanisms in somatosensation. *Annu Rev Neurosci* 24: 103–137.

Ruderman DL, Bialek W. 1994. Statistics of natural images: Scaling in the woods. *Phys Rev Lett* 73: 814–817.

Sagdullaev BT, McCall MA. 2005. Stimulus size and intensity alter fundamental receptive-field properties of mouse retinal ganglion cells in vivo. *Vis Neurosci* 22: 649–659.

Shadlen MN, Newsome WT. 1994. Noise, neural codes and cortical organization. *Curr Opin Neurobiol* 4: 569–579.

Stone C, Pinto LH. 1993. Response properties of ganglion cells in the isolated mouse retina. *Vis Neurosci* 10: 31–39.

Theunissen F, Miller JP. 1995. Temporal encoding in nervous systems: a rigorous definition. *J Comput Neurosci* 2: 149–162.

van Vreeswijk C. 2004. What is the neural code? In *23 problems in system neuroscience*, ed. J. L. van Hemmen and T. Sejnowski. New York: Oxford University Press.

van Vreeswijk, C, Somplinsky, H. 1996. Chaos in neuronal networks with balanced excitatory and inhibitory activity. *Science* 274: 1724–1726.

Victor JD. 1999. Temporal aspects of neural coding in the retina and lateral geniculate. *Network* 10: R1–R66.

Victor JD, Nirenberg S. 2008. Indices for testing neural codes. *Neural Comput* 20: 2895–2936.

Wichmann FA, Hill NJ. 2001a. The psychometric function: I. Fitting, sampling, and goodness of fit. *Percept Psychophys* 63: 1293–1313.

Wichmann FA, Hill NJ. 2001b. The psychometric function: II. Bootstrap-based confidence intervals and sampling. *Percept Psychophys* 63: 1314–1329.

Williams RW, Strom RC, Rice DS, Goldowitz D. 1996. Genetic and environmental control of variation in retinal ganglion cell number in mice. *J Neurosci* 16: 7193–7205.