Dialogue Concerning Neural Coding and Information Theory

Don H. Johnson

Computer & Information Technology Institute
Department of Electrical & Computer Engineering, MS 366
Rice University
6100 Main Street
Houston, Texas 77251–1892

June 29, 2003

Abstract

The relations between information theory and neural coding are discussed by two researchers, one knowledgeable in information theory, the other in neuroscience. The classic information-theoretic notions of entropy, mutual information, and channel capacity are clarified and possible research applications proposed.

1 Understanding basics

NS: Good morning. I wish they would get better coffee around here.

IT: Yeah, me too. How's work going these days?

NS: Pretty well. I have been exploring how information theory —something you are very familiar with—can be applied to understanding neural coding.

IT: Interesting. What kind of things are you thinking about?

NS: In neuroscience, people are using entropy and mutual information to quantify neural coding, but exactly what they are measuring and what the big picture is has lab members questioning the whole approach.

IT: Well, I'll be happy to help out if I can. But, I know nothing about neural coding.

NS: That's OK. We can each learn something. If you have time now, we can go to my office.

IT: I'm taking a break from reading e-mail. This will be much more interesting than that!

Well, information theory concerns how efficiently information sources can be compressed and transmitted. . .

NS: Hold on; that's a little too fast for me. Let me explain my situation a little.

IT: OK. I've always been intrigued by what neuroscientists like you study and, of course, I've always been curious about how the brain works.

NS: I assume you know that neurons communicate with each other with a series of pulses, each of which we call an action potential or spike. The sequence of spikes we call a spike train because they occur one after another as the boxcars on a railroad train.

IT: That's cute. It must be an old term. Anyway, I have heard that spikes occur randomly?

NS: Well, not totally random. The rate at which spikes occur for a given neuron changes with stimulus variations. More subtle changes, like in the probability distribution of the interspike intervals, can also occur. So, something about the stimulus is encoded in the pattern of action potentials but presenting the same stimulus does not result in exactly the same pattern.

IT: I understand. This kind of situation occurs in optical communication, for example, where photon arrivals are random and governed by light intensity. What's the duration of action potentials and what are the typical rates?

NS: Spikes are about a millisecond long and we have found very low discharge rates (less than 1 spike/sec and even zero) and rates as high as several thousand spikes/sec. What I mean here is not that each neuron expresses such rates; this is the range you can find as you explore various neurons throughout the brain. Cortical neurons, for example, typically range between zero and 100 spikes/s.

IT: That's pretty slow compared to modern communication systems that can operate on microsecond, even nanosecond time scales.

NS: Yes, neurons are much slower than that, but they are very effective nonetheless!

IT: I believe that! But what about the spikes represents information?

NS: Because spikes typically have the same waveform and amplitude, we assume that when spikes occur encodes information somehow.

IT: That means that spike timing is all that matters, and you have what we call a *point process channel*.

NS: That's correct. Furthermore, spike timing cannot be described as being Poisson.

IT: That makes analytic work much harder. Does spike timing deviate much from Poisson?

NS: It depends on what neuron you are recording. Auditory-nerve fibers greatly resemble Poisson processes, while other parts of the auditory pathway are far from Poisson. I would say in general that far-from-Poisson behavior is more prevalent.

IT: That's to be expected, I guess; the simple situation never occurs in the real world. Now that we have spikes and their timings, we need to worry about how they represent information.

NS: You also need to know how neurons process their inputs. Each neuron usually receives input from several neurons; depending on whether when each input produces a spike, on whether a spike is excitatory or inhibitory, and on how the neuron integrates these inputs determines the neuron's information processing function. The neuron will produce a spike only when the timing and strength of these inputs is correct. Excitatory inputs tend to cause spikes; inhibitory inputs tend to suppress them. When you consider a time interval that spans several spikes occurring in a neuron's inputs, *neural computation* amounts to each neuron producing spikes based on the timing of its inputs. And don't forget that there are lots of neurons and how they are connected to each other can be quite complicated.

IT: Wow! I thought neurons were pretty simple.

NS: The more we learn, the more we appreciate just how complicated single-neuron processing can be. I want to point out that because several inputs are processed, that means that potentially those inputs could be expressing information *together* rather than individually. We believe that neurons could make up what we call a population or an ensemble by working together to represent information. We feel that one neuron is too noisy to represent the stimulus, and that simple averaging across neurons responding to the stimulus is, well, too simple. We hypothesize that spike timings among neurons are linked somehow.

IT: Since all of these neurons are responding to the same stimulus, that would link their spike timings. I assume you mean interactions other than this?

NS: That's right. We say *stimulus-induced correlation* occurs when a common stimulus creates correlations between neurons and *connection-induced correlation* when interneuronal connections link spike timing.

IT: I see. Using technical jargon I am used to, I would call populations multi-channel point processes, which are governed by a joint probability law that does not factor (they aren't statistically independent). Furthermore, this probability law is not Poisson. Oh well, nothing like a challenging problem.

2 Entropy

NS: If need be we can make assumptions to simplify the information theory...

IT: That won't be necessary. Many results in information theory are very general. Applying them to any specific problem could be difficult but broad principles can be stated. For example, you cannot examine a signal, *any* signal like a spike train or anything else, to determine if it is carrying information or not.

NS: I thought entropy did that.

IT: That's a common confusion. Shannon proved in his momentous 1948 papers that if you wanted to represent a discrete set of objects with bits, the average number of bits/object you would need can be no smaller than the entropy of the set.

NS: What do you mean by "represent"?

IT: The idea is that you have a collection of things —objects, each of which can be distinguished from the others, that you want to identify with a bit sequence so that you can transmit which object was drawn out of a hat. The subtlety here is that rather than assigning identification numbers and simply writing them in binary (this is called simple binary coding), you can use a variable number of bits, the number of bits depending on the probability each object has of being drawn.

NS: I see. Shannon was encoding the occurrence of objects by a sequence of bits. It is kind of strange that a variable number of bits was considered. I assume you can disambiguate the bit sequences somehow.

IT: Yes, you can. But for Shannon, this is a detail. The question for him was "what is the *minimum* number of bits needed to represent the objects averaged over the probabilities with which the objects occur?" So let's assume we have N objects X_n, what information theorists call symbols, and that the objects have probabilities $p(X_n)$, $n = 1, \ldots, N$. The *entropy* of the set is given by

$$H(X) = -\sum_{n=1}^{N} p(X_n) \log p(X_n) \tag{1}$$

with the base of the logarithm taken to be two so that it units of bits. Shannon's result, today known as the *Source Coding Theorem*, says that the average number of bits needed to represent the objects *must* equal or exceed the entropy if you are going to be able to determine —decode —the object from its bit-sequence identifier. Furthermore, he showed that a decodable bit-encoding rule existed that had an average number of bits no greater than one bit larger than the entropy. His proof, though, gave no insight into what this "good" code was. A few years later, the code was discovered by a graduate student.

NS: So entropy tells me how many bits on the average I need to represent things. It must increase with the number of objects in the set, right?

IT: Yes, it does, but more importantly, it also varies with the object probabilities. You can interpret the entropy to be a measure of the complexity of the set, with the idea being that sets having larger entropies demand more bits to represent them. The worst case (largest entropy) situation occurs when the objects are equally likely to occur, and then the simple binary code works as well as anything else.

NS: OK. But doesn't entropy measure the information contained in the set of objects?

IT: Going beyond the encoding interpretation can get you in trouble. Shannon never attached meaning to the objects or to their probabilities. Information has both technical and everyday definitions. For him, an object's surprise—the negative logarithm of its probability of occurrence —was equivalent to its information. This is the technical definition needed for communication purposes, not for attaching meaning or being informative. For Shannon, an object's probabilities had an objective definition: If you did not know what they were, you could measure the symbol probabilities from data. One of his examples was the frequency of letters of the alphabet in American prose. Here, individual letters were symbols and he measured their probability of occurrence. What are these symbols in your case?

NS: Well, we chose stimuli from a finite set and study the spike trains they produce.

IT: I assume the experimenter have total control over what is presented when. Do you present stimuli randomly?

NS: Well, I don't, but others do.

IT: So a probabilistic model for your objects —the stimuli —doesn't really apply. For the others, they have *total* control over the entropy. So when you have a fixed number of stimuli, you can manipulate the entropy of the stimulus set by changing their probabilities. From what you have told me, the neuron will encode a stimulus in a way that does not depend on whether the stimuli are presented randomly or not.

NS: Well, not always. Don't forget that learning can occur, and a superficial definition of learning is that the neural response changes —adapts —somehow. That is the reason I don't present stimuli randomly; I want to make sure I can investigate sequential stimulus effects if they occur. I am trying to determine how the spike train represents the stimulus and the fidelity with which it does so.

IT: Sure, but neural encoding is almost certainly not the same as binary coding. I wish entropy would assess the coding, but it doesn't. You have the spike timings as well as the number of spikes to consider, right?

NS: That's right.

IT: Well, the number of spikes is a discrete random variable, and you could compute the entropy of that number. You would then know how many bits it would take to represent that spike count *on the average* when that number occurred in a response.

NS: Which corresponds to the randomness of that number, right?

IT: That's right. A small entropy would mean that only a select number of spikes occur in a response while a large entropy would mean that the probabilities would be more uniform.

NS: So entropy essentially measures the spread of the spike count probabilities. Is the entropy of the stimulus set related to the entropy of the spike count?

IT: Good question; let me think. You could have all stimuli produce a uniform distribution of counts; that means the output entropy is greater than the input entropy. You could also have all stimuli produce the same spike count, which has zero entropy. So, in general, there is no relation between input and output entropy.

NS: Hmmmm... What happens when you put spike timing into the mix?

IT: The story becomes more complicated because the spike timings are continuous-valued random variables. Shannon defined the entropy of a continuous random variable analogously to that of the discrete case. Letting $p_X(x)$ denote the probability density of the random variable X, its so-called *differential entropy* is defined to be

$$H(X) = -\int p_X(x) \log p_X(x)\, dx \tag{2}$$

Now, you can show that in the discrete case I wrote before, entropy is always greater than or equal to zero. In other words, the number of bits required by the Source Coding Theorem is positive.

NS: That's refreshing!

IT: However, the differential entropy can be negative! For example, the entropy of a Gaussian random variable is $\frac{1}{2}\log(2\pi e\sigma^2)$. Depending on the variance, this number can be positive, negative, even zero.

NS: It is hard to have a negative number of bits, isn't it?

IT: And that's why there is *no* Source Coding Theorem for continuous-valued sources. And it gets worse. Differential entropy depends on scaling. If you define a random variable to be a constant times another random variable ($Y = aX$), the entropy of Y is found to be related to the entropy of X as $H(Y) = H(X) + \log(|a|)$. What this means is that if you change the units of X, like express spike timings in seconds rather than milliseconds, the entropy changes!

NS: That seems weird.

IT: Correct! It can be traced to the fact that the definition of differential entropy is flawed from a physical viewpoint when data have units. In such cases, a probability density must have units of the reciprocal of the units of X. This dimensional analysis arises because $\int p_X(x)\, dx = 1$, and that "1" has no units. Well, you cannot evaluate the logarithm (or the exponential, for that matter) of something that has units. That's the flaw in the formula for differential entropy.

NS: I see. Didn't Shannon notice this?

IT: He defined it for mathematical convenience; he stated the scaling property, but never said it had physical significance.

NS: OK, so this differential entropy has utility mathematically, and the problem is interpreting what it means, right?

IT: Right, and it doesn't measure the complexity of a continuous random variable, or of a response consisting of spike timings. How do you measure the entropy of a spike train to take into account timing?

NS: Well, we always discretize the spike train. We define small time bins and assign a binary code for each bins that expresses whether a spike occurred in a bin or not.

IT: That's as I thought. There is an interesting result that says the entropy of a continuous random variable can be estimated by this kind of discretization. Let X^{Δ} denote a continuous random variables discretized to a binwidth of Δ. In the limit as Δ goes to zero,

$$\lim_{\Delta \to 0} H(X^{\Delta}) + \log \Delta = H(X)$$

NS: So the logarithm of the binwidth is kind of a correction term.

IT: But be careful if you try to correct your measured entropies this way. I worry that this correction won't work well when you compare answers obtained from different datasets using different binwidths.

NS: I understand, but you could study how the left side of your expression stabilizes as the binwidth decreases.

IT: That's right. You must be careful in estimating entropy, though; straightforward estimates have biases and you need to know whether a non-zero answer for the difference between the population entropy and the sum of individual entropies is statistically significant or not.

NS: Yes, I know. We are well aware of the statistical issues and actually have some clever algorithms for estimating entropy.

IT: Good. Also, entropy can be very useful in determining whether random variables are statistically independent or not. If you have a collection of random variables denoted by $\mathbf{X} = \{X_1, X_2, \ldots\}$, the entropy of their joint probability function equals the sum of their individual entropies *only* if they are statistically independent.

$$H(\mathbf{X}) = \sum_n H(X_n)$$

To anticipate your question, this result applies even though the actual values you get are meaningless. So, this can be used to determine if a collection has independent components.

NS: Yes, this kind of analysis has been applied to neural populations to determine if the neurons are responding independently of each other or if they are interacting with each other to produce some kind of combined response. When we find the entropy of a population response to be larger than the sum of its component entropies, we call that a *synergistic* response, the idea being that the neuron's interactions result in more information being expressed by the population than by considering them individually. From what you say, that interpretation could be suspect.

IT: That's right. The interpretation may not be on target, but using entropy to determine whether the neural responses are statistically independent or not is OK. However, synergy as you have defined it cannot occur. An elementary property of the entropy of a collection of random variables is that it *must* be less than the sum of individual entropies.

$$H(\mathbf{X}) \leq \sum_{n=1}^{N} H(X_n)$$

NS: I didn't know that. We also use mutual information to assess synergy.

IT: OK. That might work. But getting back to neurons, you want to assess neural coding?

3 Quantifying information processing

NS: One of the applications we have for information theory is measuring how much information "gets through" a neuron. We use the mutual information between the stimulus and the response. I am unsure, however, how to interpret the answers we get.

IT: Shannon invented mutual information, but it actually doesn't come up in communications problems that often except in one *very* important situation. If X and Y are the input and output of a communications channel, the mutual information between them is defined to be

$$I(X;Y) = \int p_{X,Y}(x,y) \log \frac{p_{X,Y}(x,y)}{p_X(x)p_Y(y)}\,dxdy = \int p_{Y|X}(y|x)p_X(x) \log \frac{p_{Y|X}(y|x)}{p_Y(y)}\,dxdy \qquad (3)$$

As with entropy, we usually use base-2 logarithms so that the answer has units of bits.

NS: So how do you interpret it?

IT: Well, it equals zero when the numerator and denominator in the argument of the logarithm are equal. This means that when the input and output are statistically independent of each other, the mutual information is zero. When the output depends on the input, the mutual information is always positive. The more the output Y reflects the input, the greater the mutual information. The upper limit depends on whether you have discrete random variables or not.

NS: Uh oh. I suspect that the continuous random variable case causes problems.

IT: Well, not really. The two cases are just different. If you have a discrete random variable, the upper limit occurs when $Y = X$ and in this case the mutual information is the entropy of if the input $H(X)$. By the way, the integral in my expression for the mutual information is a sum in the discrete case.

NS: I assumed that. But what about the continuous random variable case?

IT: Again, the maximum occurs when the output equals the input, and the maximum is infinity. You can derive this by considering two jointly Gaussian random variables and letting the correlation coefficient approach one.

NS: I'll try to work that out later. But this difference between the discrete- and continuous-random variable cases seems bizarre. Why should the answers be so different?

IT: I don't think they are. It is this result that argues that the entropy of a continuous quantity is indeed infinite. From the perspective of the Source Coding Theorem, this result would mean that it takes an infinite number of bits to represent a continuous quantity without introducing error. That actually sounds correct to me.

NS: That makes sense to me too. But what does mutual information mean?

IT: At a fundamental level, the mutual information measures how similar the input and output are. They are maximally similar when they are equal and that's when the mutual information is the largest. From a communications viewpoint, the mutual information expresses how much the output resembles the input. Note that the mutual information expression can be manipulated to obtain

$$I(X;Y) = H(Y) - H(Y|X)$$

The last term is known as the conditional entropy and is given by

$$H(Y|X) = -\sum_{x,y} p_{X,Y}(x,y) \log p_{Y|X}(y|x)$$

Because this term is an entropy, it measures how "random" the *conditional* probability distribution of the output is, on the average, given a specific input. The more random it is, the larger the entropy, which reduces the mutual information. The less random it is, the smaller the entropy until it equals zero when $Y = X$.

NS: So it does measure how much of the input appears in the output?

IT: Yes, it does.

NS: Any measurement problem when you discretize a spike train?

IT: Other than the usual caveat about determining when the binwidth is small enough, no correction term is needed. The correction terms cancel even when the input and output have different units.

NS: That's good. But let me tell you how we use mutual information. The input is the stimulus, so let me denote it S. The output is the response R of a neuron. Because the response is random and varies with time after the stimulus is presented, we essentially estimate how the response probability varies with time by averaging over several presentations of the same stimulus.

IT: OK. Mathematically what you just said is that you have estimated the conditional probability $p(R|S)$, where the response random variable actually consists of the response at several times. We would call that a random vector. You estimate this using what we call ensemble averaging. That's really a detail, but helps me think about what you are doing. What you need for a mutual information calculation is $p(R, S)$, which can be found from the simple formula $p(R|S)p(S)$. But what is $p(S)$?

NS: To measure mutual information, we present the stimuli randomly so that we can define a probability $p(S)$ and then…

IT: Hold on! You said you don't like to present stimuli randomly. So you change the way you present stimuli to quantify neural information processing? Let's suppose there are no sequential effects of one stimulus changing the way the neuron responds to another stimulus. Shouldn't you get the same response characteristics as you did when you present them randomly?

NS: Yes, but we want to calculate mutual information so that we can characterize how much of the input information gets through to the neuron's response.

IT: That's not what it is going to do. Suppose I add another stimulus to the mix. Will the neuron's response change (clairvoyantly almost) to take into account the fact that you have changed the stimulus set?

NS: Well, probably not.

IT: Well, you realize the mutual information will change just because you change the number of stimuli or you change the stimulus probabilities.

NS: Why is that?

IT: Remember that the maximum value of mutual information is the entropy of the stimulus set. Consequently, even though the neuron's response characteristics aren't changing, the mutual information can change.

NS: Hmmm. Suppose I search over stimulus probabilities and number of stimuli until I find the maximum mutual information. Wouldn't that tell me something about the neuron?

IT: Wow! You just hit upon one of the central concepts in information theory.

4 Channel capacity

IT: In Shannon's original paper, he produced a second result much more astounding than the Source Coding Theorem. Today, it is known as the Noisy Channel Coding Theorem. In digital communication, the bottom line is transmitting data as fast as you can without incurring too many errors. Shannon showed that as long as the datarate is less than the *channel capacity*, digital transmission is possible with *no* error. Furthermore, if you transmit with a datarate greater than capacity, the errors will be overwhelming.

NS: Sounds like something magic happens for datarates near capacity.

IT: Absolutely; there is a wall at capacity that separates error-free from error-prone communication. Underlying all of this is the idea of error correcting codes. For example, simply sending the same bit several times allows you to figure out what the actual data bit was so long as there aren't too many errors. What Shannon was saying implicitly is that so long as the datarate is less than the capacity, there exists an error correcting code than can correct *all* errors that might occur in the transmission process.

NS: What does this all have to do with mutual information?

IT: The underlying model is that the data and error correcting bits pass through a digital channel that creates errors. Because of the channel...

NS: Just a second. What do you mean by "channel"?

IT: In communications, the channel is a system that describes what happens to a signal in the transmission/reception process. Rather than describing just the communication apparatus, channel models reflect the physical effects of transmitting information. Channels add noise, distort, and generally make the receiver work hard to recover the transmitted signal with as high a fidelity as possible. As I like to put it, nothing good happens in a channel, and the transmitter and the receiver are designed together to best mitigate the channel's effects on communication.

NS: Sorry for the interruption, but channels for us are very different. The ions flow through channels in the neuron membrane to create the voltage changes we measure. We like channels.

IT: Actually, I like communication channels. If it weren't for the trouble they cause, communications problems would have been solved a long time ago. But let us get back to digital communications.

As I said, the digital channel causes reception errors to occur. However, if the datarate is less than the channel capacity, the receiver can determine what bits were received in error and correct the errors. Thus, the bits going into the channel comprise the input X and the received data bits (before they are corrected) is the output Y. The capacity is the maximum of the mutual information $I(X;Y)$ with respect to the probability function of the input $p(X)$.

$$C = \max_{p(X)} I(X;Y) \tag{4}$$

So what you said about maximizing the mutual information with respect to number of stimuli and their probabilities essentially means you are finding the neural system's capacity. Because of the expression I wrote earlier for the mutual information (equation (3)), maximizing with respect to p_X results in capacity depending only on $p_{Y|X}(y|x)$, which defines how the output changes with the input.

NS: That means our use of mutual information to characterize how a neural system processes information is not off the mark.

IT: That's right. But be forewarned not to get too excited about the input probability distribution that achieves the maximum mutual information in the capacity definition. In all cases I know of, the maximizing probability distribution cannot represent information.

NS: Why is that?

IT: Well, the simplest example is called the binary symmetric channel. Here, zeros and ones are transmitted through a channel that "flips" bits with some probability. The maximizing probability distribution of ones and zeros is for the bits to transmitted statistically independently and to have equal probability. I would call this binary-valued white noise. I can't conceive what use this kind of signal has for representing information.

NS: So if the probabilities of one and zeros change according to some input, you no longer have the capacity-achieving distribution.

IT: Right. Anyway, from what you said, I don't think neurons serve as digital communication systems.

NS: Well, exactly what do you mean?

IT: In digital communications, you have a discrete set of things to transmit from one place to another. You represent each of them by a unique sequence of bits. Each bit is in turn represent by a signal, and one of these signals is transmitted through a communications channel that introduces noise and other disturbances. The receiver's job is to figure out from the signal it receives which bit was sent and from the received bit sequence what "thing" was being transmitted.

NS: Well, we do use a set of stimuli in our experiments.

IT: But aren't they chosen by selecting a certain set of parameter values, which in the real world vary continuously? I have in mind the brightness of light, for example.

NS: That's true. But you said mutual information did not depend on the discretization!

IT: Yes, I did. Let me be clearer. The assumption here is that you quantize some continuous variable and the probability of the each bin occurring equals the integral of the probability distribution over that bin. I'll bet you aren't discretizing your stimuli by binning. Sounds to me more like sampling.

NS: If you mean we select the stimuli we present rather than averaging them somehow, you are correct.

IT: In that case, the asymptotic invariance of mutual information to binwidth does *not* apply.

NS: OK, but that's a measurement problem. Wouldn't a capacity calculation still be interesting?

IT: Yes, but it is complicated. The assumption underlying the Noisy Channel Coding Theorem is that the input is ultimately discrete. For example, a digital source might be the sequence of alphabetic characters representing a novel. The Theorem does *not* apply in this form to communicating analog signals; *analog signals cannot be communicated through a noisy channel without incurring error, no matter what the transmission scheme.*

NS: I have more questions about capacity, though. We have been computing the information contained in a spike train by maximizing mutual information in the way I described. I guess our measurements obtained by sampling stimuli are flawed. Suppose we did actually discretize the stimulus by a binning process somehow. If we kept constant the stimulus amplitude bin size and the temporal binwidth for the spike train and manipulated the input probabilities and made them small enough, wouldn't we still be approximating the actual capacity?

IT: Yes, but did you know that information theorists long ago analytically derived the capacity of the point process channel?

NS: Really! You mean we don't need to measure a neuron's capacity?

IT: That's right as long as the neuron's response is well-described as a point process. First of all, suppose the *instantaneous* rate at which spikes occur cannot be lower than $\lambda_{\min}$ nor greater than $\lambda_{\max}$. Furthermore, impose the constraint that the average rate cannot be larger than $\bar{\lambda}$.

NS: I could measure these for a given neuron by applying a stimulus, right?

IT: That's right.

NS: How about if I stimulate the neuron artificially by directly injecting current into it and measuring these rates?

IT: If you want the single-neuron capacity for a stimulus, you need the stimulus-driven numbers. I would think you would want to measure what the stimulus evokes. Also, let me stress that these maximal and minimal rates are obtained with *dynamic* stimuli.

NS: What?

IT: The $\lambda_{\min}$ and $\lambda_{\max}$ parameters are the achievable limits of the neuron's instantaneous rate response. The average rate parameter $\bar{\lambda}$ is the maximal sustained rate at which the neuron can produce spikes.

NS: Some neurons are spontaneously active when no stimulus is applied, which would make $\lambda_{\min}$ that spontaneous rate, right?

IT: As long as the neuron's response to *any* stimulus does not go below that rate.

NS: Well, usually there are transitory responses than can dip below the spontaneous rate.

IT: $\lambda_{\min}$ is that minimal response rate.

NS: Interesting. So what determines capacity are the limits to which the neurons can respond transiently and over the long haul.

IT: That's right. Information theorists have shown that channel capacity is achieved by a Poisson process having an event rate that bounces randomly between minimum and maximum rates ($\lambda_{\min}$ and $\lambda_{\max}$ respectively), and equals

$$C = \frac{1}{\ln 2}\begin{cases} \lambda_{\min}\left[\frac{1}{e}\left(\frac{\lambda_{\max}}{\lambda_{\min}}\right)^{\lambda_{\max}/(\lambda_{\max}-\lambda_{\min})} - \ln\left(\frac{\lambda_{\max}}{\lambda_{\min}}\right)^{\lambda_{\max}/(\lambda_{\max}-\lambda_{\min})}\right], & \bar{\lambda} > \lambda^\circ \\ (\bar{\lambda}-\lambda_{\min})\ln\left(\frac{\lambda_{\max}}{\lambda_{\min}}\right)^{\lambda_{\max}/(\lambda_{\max}-\lambda_{\min})} - \bar{\lambda}\ln\frac{\bar{\lambda}}{\lambda_{\min}}, & \bar{\lambda} < \lambda^\circ, \end{cases}$$

where

$$\lambda^\circ = \frac{\lambda_{\min}}{e}\left(\frac{\lambda_{\max}}{\lambda_{\min}}\right)^{\lambda_{\max}/(\lambda_{\max}-\lambda_{\min})}.$$

I know the expression is complicated, but this is it! By the way, the curious division by $\ln 2$ makes this result have units of bits/s. Also, the rate signal that achieves capacity swings between the minimum and maximum rates in a square-wave fashion randomly with skewed probabilities: the ratio of the probabilities of having maximum and minimum rates is $(\lambda_{\max}-\lambda^\circ)/(\lambda^\circ-\lambda_{\min})$. Hardly a signal that could be used for communicating anything.

NS: That is indeed wierd, but I am startled that we need not estimate the capacity for a single neuron? Does this result apply to non-Poisson models, which much more accurately describe spike trains?

IT: Yes. Don't be misled by the Poisson process part of the result. Channel capacity is always the result of maximizing the mutual information. What I said means that the Poisson process fell out of the maximization process. All other point process models produced a mutual information less than or equal to that of the Poisson.

NS: So, more realistic models may have a smaller capacity?

IT: It was explicitly shown that non-Poisson point processes cannot have larger capacities. But don't ask which ones do and which ones don't. These capacity derivations are exceedingly difficult.

NS: What a minute. I just noticed something. Your definition of capacity (equation 4) is maximal mutual information, which by equation (3) has units of bits. Your equation for capacity has units of bits/s. How can that be?

IT: Good point, but nothing mysterious is occurring. For channels through which you send signals that depend on continuous time, like a neuron, the definition for capacity is slightly different.

$$C = \lim_{T\to\infty}\frac{1}{T}\max_{p(X)} I(X;Y)$$

Here, T denotes the time interval during which communication occurs. This definition of capacity has units of bits/s.

NS: OK, I understand. It would be interesting to compare the analytic formula with what we measure. We express our results in bits/spike.

IT: That's an interesting way to look at it. I would compute that by dividing the capacity by the average rate $\bar{\lambda}$.

NS: Let's work an example. Let's say the minimum rate is zero and the maximum rate is 100 spikes/s with an average rate of 25 spikes/s.

IT: OK, but first of all, when $\lambda_{\min} = 0$, the formula simplifies greatly.

$$C = \frac{1}{\ln 2} \begin{cases} \dfrac{\lambda_{\max}}{e}, & \bar{\lambda} > \lambda^{\circ} \\ \bar{\lambda} \ln\left(\dfrac{\lambda_{\max}}{\bar{\lambda}}\right), & \bar{\lambda} < \lambda^{\circ} \end{cases}$$

and $\lambda^{\circ} = \lambda_{\max}/e = 0.37\lambda_{\max}$. Since you chose an average rate less than $0.37\lambda_{\max}$, the answer, in units of bits/spike, is $\ln(\lambda_{\max}/\bar{\lambda}) / \ln 2 = 2$ bits/spike.

NS: That's about the kinds of values we get!

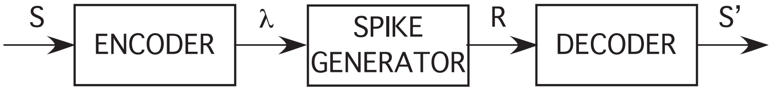
IT: Yes, but I don't know if this answer means that much. It does *not* mean that each spike represents 2 bits worth of information. Don't forget that the capacity result applies to *digital* communication, wherein you would be using a digital coding scheme (representing bits with specific rate variations) that would have certain values for $\lambda_{\min}$, $\lambda_{\max}$, and $\bar{\lambda}$. Within these rate restrictions, the capacity reveals the maximum datarate that digital data could be sent by a point process channel without incurring massive transmission errors. The signals representing the bits would be the rate of discharge, and they would vary between minimum and maximum rates.

NS: I see. We certainly don't measure responses that look like square-wave rate changes. In general, spike rate varies in complicated ways when a stimulus is presented.

IT: That's why I am sure you don't have a digital communications problem; what you want is the *analog* version of the Noisy Channel Coding Theorem.

NS: It at least sounds more appropriate; what is it?

IT: Let's assume the stimulus is encoded as a rate of discharge, that rate produces a measured response, and this response is decoded into some internal representation that ultimately we can use to compare with the input stimulus. Does this model make sense?



NS: Well, it is superficial; it doesn't tell me anything new.

IT: Well, that's the point; we want a generic model that applies broadly.

NS: That's OK then.

IT: First of all, we define a distortion measure $d(S, S')$ between the input, which I would take to be the stimulus S, and its value S' after it has been neurally encoded and decoded. The distortion measure quantifies how important dissimilarities between the original and decoded signals are. For example, $d(S, S')$ could be $(S - S')^2$, in which case we are saying that the squared error quantifies the encoder/decoder system's performance. We again assume the stimulus is stochastic and is described by a probability function. Because the coding and decoding operations are noisy, the decoded stimulus is also stochastic. By the way, S does not represent one value for stimulus parameters; it is intended to represent how the stimulus varies over time, space and any other relevant stimulus dimension.

NS: So you mean S represents all possible stimuli that could be presented?

IT: That's right. And S' denotes the decoded stimulus.

NS: If I only know what that was... Sorry; daydreaming. Continue.

IT: We can compute the *average distortion* D by averaging the distortion measure over all possible stimulus-decoded stimulus pairs.

$$D = \iint d(S,S')p(S,S')\,dS dS'$$

Here, $p(S,S')$ is the joint distribution between the stimulus and the decoded stimulus. Finding the average distortion requires a specification of the stimulus probability distribution $p(S)$ and sending stimuli according to that probability law through the sensory system. The conditional distribution $p(S\,'|S)$ determines the statistical law governing the encoding and decoding systems. In your case, the encoding and decoding are fixed.

NS: But I don't know what the probability law of the encoding/decoding system is. And I don't know what the distortion function should be. Should I use a perceptual distortion measure or something arbitrary? I guess I am asking if it matters what I choose for the distortion.

IT: You bet it does! To apply the Noisy Channel Coding Theorem to a real system, you should analyze it using the guidelines that governed its "design" (it's the engineer in me talking here).

NS: But that's what we are trying to discover!

IT: I realize we have a chicken-and-egg problem, but let's go on.

NS: Alright, but so far this result doesn't look very useful.

IT: We'll see. Shannon then defines $\mathcal{R}$ to be the rate at which information can be reproduced to a given distortion D_0 using the mutual information between the encoded stimulus and the response it produces.

$$\mathcal{R} = \min_{p(S,S')} I(\lambda; R) \text{ with } D \leq D_0$$

Shannon refers to it as a rate despite it having units of bits.

NS: I want to be clear here; what's the difference between the encoded stimulus λ and the response R?

IT: The encoding is how the rate of discharge and interspike timing of a single neuron or the combined discharge probabilities for a population depend on the stimulus. The response is the actual sequence of action potentials produced by this encoding.

NS: OK, I have them straight. But what is this rate quantity $\mathcal{R}$?

IT: The decoded output S' is derived from the response R by the decoder. There is a "rule" relating the stimulus to the encoding λ and a rule relating the response to the decoded output. These could have stochastic elements, but let's keep it simple for now. When all is said and done, the stimulus and the output are related to each other somehow, and this relationship is described by the conditional probability function $p(S\,'|S)$. The joint distribution we need for the average distortion calculation is $p(S'|S)p(S)$. You now think about the stimulus (through its probability function) varying over all possible stimuli. Some stimuli, after being encoded, producing a response, and being decoded yield a large distortion; others are more easily encoded and decoded, thereby yielding a smaller distortion. Recall than mutual information $I(\lambda; R)$ is the smallest when the encoding and the response are statistically independent of each other, which *will* produce a large distortion. So the minimization corresponds to finding the worst-case stimulus probability distribution that yields an average distortion that meets your specification D_0.

NS: That I'll believe. I take it that D_0 is some threshold distortion value that is acceptable. Where does that come from?

IT: You choose it.

NS: This is all quite complicated, but let me attempt a succinct summary. This rate $\mathcal{R}$ is found by searching for hard-to-code stimuli that just pass my distortion test, and the mutual information measures how good the translation from an encoding into a response has to be to create a just-acceptable distortion.

IT: That's very good. The analog Noisy Channel Coding Theorem states that so long as $\mathcal{R} < C$, where capacity is computed by maximizing the mutual information $I(\lambda; R)$ with respect to the probability function of λ, the stimulus can be encoded in such a way that the required distortion criterion is met. The more stringent the criterion (the smaller D_0), the larger $\mathcal{R}$ becomes until one cannot send information that way and meet the distortion criterion.

NS: But I don't know what distortion measure to apply and I don't know a threshold value.

IT: Not knowing the measure is difficult to get around; you just have to chose one. Information theorists plot the rate $\mathcal{R}$ as a function of the threshold, so you can study that behavior rather than having to select a specific threshold value.

NS: How about the capacity? How do I find it?

IT: For a single neuron, it is the capacity I gave you for the point process channel. So that's easy. If you are talking about a neural population, I don't know what the capacity is. If the population components were statistically independent of each other, the population's capacity would equal the sum of the component capacities, but from what you have said, this result may not be all that useful.

The difficult quantity to find is the rate $\mathcal{R}$; the constrained minimization with respect to the joint probability function $p(S, S')$ can be quite difficult to compute, even if you just need to vary $p(S)$. The reason is that finding the joint probability function requires you to specify how the entire stimulus reconstruction system works, then find the joint distribution of its input and output.

NS: So I have my 2 bits/spike for the capacity, let's say. Then, information rates $\mathcal{R}$ less than this exist that can result in a reconstructed stimulus that is guaranteed to meet some distortion criterion. I guess I could search over some hypothesized encoders and decoders to see if the resulting rate exceeds my capacity. If they do, they don't exist; if they don't exceed my capacity, they are possible.

IT: That approach would work if it weren't for the threshold distortion D_0. Basically, anything works as the threshold increases, and fewer work as it is lowered. And again, you must specify the distortion function $d(S, S')$.

NS: That may not be as hard as you think. We have lots of psychophysical results that describe how humans perform on perceptual tasks. Quite possibly we could find a credible distortion function that fits the data.

IT: That does sound promising, but I cannot stress how difficult finding the joint distribution of the input stimulus and the decoded stimulus will be. And then you have to search over all such joint distribution functions.

NS: Consider me forewarned; doesn't mean it is not worth trying.

IT: Agreed. Especially if it winds up revealing something interesting.

5 Conclusions

NS: This has been quite illuminating; I'm glad I dropped by. So I know that we can't really measure how much information a signal is conveying by using entropy, and that I have been misinterpreting the capacity numbers. However, all is not lost; information theory might provide some help in understanding the brain.

IT: Well, I am not sure. Warren Weaver in his prologue to the book version of Shannon's paper reminds us that Shannon's work was entitled "A Mathematical Theory of Communication." Its powerful results concern the limits of effective communication of signals, and said little about what was being communicated. For example, as we have discovered, trying to discern the information carried by a signal is not addressed by what has become "information theory." Weaver said that future research must be concerned with semantics: determining what the information is and how best to determine the intended meaning. Information theorists know that there are many open problems.

NS: Well, I happen to think that the brain is the best proving ground for any results the information theory community comes up with.

IT: I think you're right. Let's keep working together.

NS: OK. Be happy to.

Acknowledgements

This work was supported by grants from the National Institute of Mental Health and from the National Science Foundation.

Readings

Berger, T. (1971). *Rate Distortion Theory*. Prentice-Hall, Englewood Cliffs, NJ.

Brémaud, P. (1981). *Point Processes and Queues*. Springer-Verlag, New York.

Cover, T. and Thomas, J. (1991). *Elements of Information Theory*. Wiley, New York.

Davis, M. (1980). Capacity and cutoff rate for Poisson-type channels. *IEEE Trans. Info. Th.*, IT-26:710–715.

Johnson, D. (1996). Point process models of single-neuron discharges. *J. Comp. Neuroscience*, 3:275–299.

Kabanov, Y. (1978). The capacity of a channel of the Poisson type. *Theory Prob. and Applications*, 23:143–147.

Shannon, C. (1948). A mathematical theory of communication. *Bell System Tech. J.*, pages 379–423, 623–656.

Shannon, C. and Weaver, W. (1949). *The Mathematical Theory of Communication*. U. Illinois Press, Urbana, IL.