MICHALE FEE: OK, so we're going to start a new topic today. We're going to spend the next three or so lectures talking about spectral analysis. And we're going to warm up to that topic today by talking about time series more generally.

Now, one of the things that we're going to discuss in the context of this new topic-- so I'm going to spend a few minutes reviewing a little bit about receptive fields that we've talked about in the last lecture. And one of the really cool things that I find in developing tools for analyzing data is that there's a really big sense in which, when we developed tools to analyze data, we're actually developing tools that kind of look like what the brain does.

So our brains basically learn to analyze sensory stimuli and extract information from those sensory stimuli. And so when we think about developing tools for analyzing data, we take a lot of inspiration from how neurons and brain circuits actually do the same thing. And a lot of the sort of formulation that we've developed for understanding how neurons respond to sensory inputs has a lot of similarity to the kind of things we do to analyze data.

All right, so a brief review of mathematical models of receptive fields-- so the basic, most common model for thinking about how neurons respond to sensory stimuli is the linear/non-linear model. And again, the basic idea is that we have a sensory stimulus. In this case, this is the intensity of a visual field. So it's intensity as a function of position x and y-- let's say on the screen or on a retina. Then that stimulus goes through a filter. And the filter is basically a pattern of sensitivity of the neuron to the sensory input.

And so in this case, I've represented this filter as a filter that's sensitive to a ring of light around a center of darkness. So this might be like an off neuron in the retina.

So that filter acts on the stimulus. It filters some aspect of the stimulus, and develops a response to the stimulus. That response goes to what's called an output non-linearity, which typically looks something like this, where a very negative response produces no spiking of the neuron, no output of the neuron, whereas a

large overlap of the stimulus with the filter, with the receptive field, produces a large spiking response.

So a typical way this would look for a neuron is that if the filter response, L, is 0, the neuron might have some spontaneous firing rate, r0. And the firing rate of the neuron is modulated linearly around that spontaneous firing rate r by an amount proportional to the response of the filter. And then obviously if the response of the filter is very negative, then the firing rate of the neuron at some point reaches 0. And if the r0 plus L goes below 0, then the firing rate of the neuron can obviously not go negative. And so the firing rate of the neuron will just kind of sit at that floor of 0 firing rate.

All right, so that is a response-- that is an output non-linearity. And then most neurons fire sort of randomly, at a rate corresponding to this firing rate that is the output of this output nonlinearity.

And so what happens is a neuron generates spikes probabilistically at a rate corresponding to the output of this non-linear response function. OK, any questions about that? All right, so what we're going to do today is I'm going to take a little bit of a detour and talk about how we think about the randomness or the stochasticity of neuronal firing rates. OK, and I'll talk about the Poisson process. And then we're going to come back and think about filters more generally, and how we can analyze signals by applying filters of different types to them.

OK, so I think this is basically what we covered last time. Again, the idea is that we can think of the response of a neuron as a spontaneous firing rate plus a filter acting on a stimulus input. In this case, the filter is a two-dimensional filter. So here I'm just fleshing out what this looks like here, for the case of a linear filter in the visual system, a spatial receptive field. So G is the spatial receptive field. i is the intensity as a function of position.

And what we do is we multiply that spatial receptive field times the stimulus, and integrate over all the spatial dimensions x and y. In one dimension, we would have a spatial receptive field that looks like this. So this receptive field is sensitive to a positive brightness in the center, and a negative or a dark feature in the surrounding area. And again, the way we think about this is that the neuron is

maximally responsive if the pattern of sensory input looks like the receptive field, is highly correlated with the receptive field. So if the receptive field has a positive central region surrounded by negative flanks, then that neuron is maximally responsive if the pattern of light looks like the receptive field.

So if the light pattern has a bright spot surrounded by dark flanking regions, then we calculate this integral, what you find is that the positive parts-- the positive receptive field times the positive intensity or brightness multiplies to give you a positive contribution to the neuronal response. A negative component of the receptive field multiplies by a negative component of the intensity. And that gives you a positive contribution to the response. And so you can see that even though the receptive field has positive and negative parts, so does the intensity function have positive and negative parts. And when you multiply those together, you get a positive contribution to the response of the neuron everywhere. And so when you integrate that, you get a big response.

In contrast, if the intensity profile looked like this-- it's very broad. So this looks like a bright spot surrounded by a dark ring. If, on the other hand, you have a large bright spot that completely overlaps this receptive field, then when you multiply these two functions together, this positive times positive will give you a positive here. But the negative part of the receptive field overlaps with a positive part of the intensity. And that gives you a negative contribution to the neuronal response. And when you integrate that, the positive here is canceled by the negative there, and you get a small response.

All right, any questions about that? I think we covered that in a lot of detail last time. But again, the important point here is that this neuron is looking for a particular kind of pattern in the sensory input. And it responds when the sensory input has that pattern. It doesn't respond as well when the sensory input has a different pattern.

And we have the same kind of situation for the sensitivity of neurons to temporal patterns. So we can write down the firing rate of a neuron as a function of time. It's just a spontaneous firing rate plus a filter acting on a time-dependent stimulus.

So in this case, this filter will be looking for or sensitive to a particular temporal

pattern. And as you recall, if we have a time-dependent stimulus-- let's say this is the intensity of a spot of light, that you can have a neuron that's responsive to a particular temporal pattern. Let's say a brief darkening of the stimulus followed by a pulse of bright high intensity, and then the neuron response after it sees that pattern in the stimulus.

And the way we think about this mathematically is that what's happening is that the stimulus is being convolved with this linear temporal kernel. And the way we think about that is that the kernel is sliding across the stimulus. We're doing that same kind of overlap. We're multiplying the stimulus times the kernel, integrating over time, and asking, where in time does the stimulus have a strong overlap with the kernel? And you can see that in this case, there's a strong overlap at this point. The stimulus looks like the kernel. The positive parts of the stimulus overlap with positive parts of the kernel. Negative parts of the stimulus overlap with negative parts of the kernel.

So when you multiply that all together, you get a big positive response. And if you actually slide that across and do that integral as a function of time, you can see that this convolution has a peak at the point where that kernel overlaps with the stimulus. And right there is where the neuron would tend to produce a spike.

All right, and so, I think near the end of the last lecture, we talked about integrating or putting together the spatial and temporal parts of a receptive field into sort of a larger concept of a spatio-temporal receptive field that combines both spatial and temporal information.

All right, so here are the things that we're going to talk about today. We're going to again, take a little bit of a detour, and talk about spike trains being probabilistic. We'll talk about a Poisson process, which is the kind of random process that most people think about when you talk about spike trains of neurons. We're going to develop a couple of measures of spike train variability.

So an important thing that neuroscientists often think about when you measure spike trains is how variable are they, how reproducible are they in responding to a stimulus. And a number of different statistical measures have been developed to quantify spike trains. And we're just going to describe those briefly. And I think you'll

have a problem set problem that deals with those.

And then I'm going to come back to kind of a broader discussion of convolution. I'll introduce two new metrics or methods for analyzing time series data, data that's a function of time. Those are cross-correlation and autocorrelation functions. And I'm going to relate those to the convolution that you've been using more often and we've been seeing in class.

And then finally we're going to jump right into spectral analysis of time series, which is a way of pulling out periodic signals from data. And what you're going to see is that that method of pulling out temporal structure out of signals looks a lot like the way we've been talking about how neurons have sensitivity to temporal structure in signals. OK, we're going to use that same idea of taking a signal and asking how much does it overlap with a linear kernel with a filter. And we're going to talk about the kind of filters you use to detect periodic structure and signal. And not surprisingly, those are going to be periodic filters.

All right so that's what we're going to talk about today. All right, so let's start with probabilistic spike trains. So the first thing that you discover when you record from neurons in the brain and you present a stimulus to the animal-- let's say you record from neurons in visual cortex or auditory cortex, and you present a stimulus for some period of time, what you find is that the neurons respond. They respond with some temporal structure. But each time you present the stimulus, the response of the neuron is a little bit different.

So you can see that this-- so what I'm showing here is a raster plot. So each row of this shows the spiking activity of a neuron during a presentation of this stimulus. The stimulus is a bunch of dots presented that move across the screen. And this is a part of the brain that sensitive to movement of visual stimuli. And what you can see is that each time the stimulus is presented, the neuron generates spikes.

Each row here is a different presentation of the stimulus. If you average across all of those rows, you can see that there is some repeatable structure. So the neuron tends to spike most often at certain times after the presentation of this stimulus. But each time the stimulus is presented, the spikes don't occur in exactly the same place. So you have this sense that when you present the stimulus, there is some sort

of underlying modulation of the firing rate of the neuron. But the response isn't exactly the same each time. There's some randomness about it.

So we're going to talk a little bit about how you characterize that randomness. And the way that most people think about the random spiking of neurons is that there is a-- sorry about that. That mu was supposed to be up there.

So let's go to a very simple case, where we turn on a stimulus. And instead of having a kind of a time-varying rate, let's imagine that the stimulus just comes on, and the neuron starts to spike at a constant average rate. And let's call that average rate mu.

So what does that mean? What that means is that if you were to present this stimulus many, many times and look at where the spikes occur, there would be some uniform probability per unit of time that spikes would occur anywhere under that stimulus during the presentation of that stimulus.

So let's break that time window up during the presentation of the stimulus into little tiny bins, delta-T. Now, if these spikes occur randomly, then they're generated independently of any other spikes, with an equal probability in each bin. And what that means is that if the bins are small enough, most of the bins will have zero spikes.

And you can write down the probability that a spike occurs in any one bin is the number of spikes per unit time, which is the average firing rate, times the width of that bin in time. Does that make sense? The probability that you have a spike in any one of those very tiny bins is just going to be the spikes per unit of time times the width of the bin in time.

Now, that's only true if delta-T is very small. Because if delta-T gets big, then you have some probability that you could have two or three spikes. And so this is only true in the case where delta-t is very small. So the probability that no spikes occur is 1 minus mu delta-T. And we can ask, how many spikes land in this interval T? And we're going to call that probability P. And it's the probability that n spikes land in this interval, T.

And we can calculate that probability as follows. So that probability is just the

product of three different things. It's the probability of having n bins with a spike. So that's mu delta-T to the n. It's n independent events, with probability mu delta-T, times the probability of having M minus n beans with no spike. So that's 1 minus mu delta-T to the M minus n. And we also have to multiply by the number of different ways that you can distribute those n spikes in M bins. And that's called M choose n. Yes.

AUDIENCE: So when we pick delta-T, we still have to pick it big enough so that it's not less than how long the [INAUDIBLE], right? Because you can't have--

MICHALE FEE: Yeah, so, OK, good question. So just to clarify, we're kind of imagining that spikes are delta functions now. So we are-- so in this case, we're imagining that spikes are not like produced by influx of sodium and an outflux of potassium, and it takes a millisecond.

In general, spikes are, let's say, a millisecond across. And we're usually thinking of these bins as kind of approaching about a millisecond. But if you-- what we're about to do, actually, is take the limit where delta-T goes to 0. And in that case, you have to think of the spikes as delta functions. OK, so the probability that you have n spikes in this interval, T, is just the product of those things. It's the probability of having n bins with a spike times the number of different ways that you can put n spikes into M bins.

So you multiply those things together, and you take the limit that delta-T goes to 0. And it's kind of a cute little derivation. I've put the full derivation at the end so that we don't have to go through it in class. But it's kind of fun to look at anyway. And what you find is that in the limit, that delta-T goes to 0. Of course as delta-T goes to 0, the number of bins goes to infinity. Because the number of bins is just capital T divided by delta-T.

So you can go through each of those terms and calculate what happens to them in the limit that delta-T goes to 0. And what you find is that the probability of having n spikes in that window T just like mu T to the n. What is mu T? mu T is the expected number of spikes in that interval. It's just the number of spikes per unit time, times the length of the window divided by n factorial, times e to the minus mu T. And again, mu T is the expected number of spikes. And that is the Poisson distribution.

And it comes from a very simple assumption, which is just that spikes occur. The spikes occur independently of each other, at a rate mu spikes per second, with some constant probability per unit time of having a spike in each little time bin.

Now notice that if you have a rate-- there's some tiny probability per unit of time of having a spike there, probability of having a spike there, probability of having a spike there, and so on-- you're going to end up sometimes with one spike in the window, sometimes with two spikes, sometimes with three, sometimes four, sometimes five. If this window's really short, you're going to have more cases where you have zero or one spikes. If this window is really long, then it's going to be pretty rare to have just 0 or 1 spikes. And you're going to end up with, on average, 20 spikes, let's say.

So you could see that, first of all, the number of spikes you get is random. And it depends on the size of the window. And the average number of bikes you get depends on the size of the window and the average firing rate of the neuron. OK, so let's just take a look at what this function looks like for different expected spike counts.

So here's what that looks like. So we can calculate the expected number of spikes. And just to convince you, if we calculate the average number of spikes using that distribution, we just sum, over all possible number of spikes, n times the probability of having n spikes. And when you do that, what you find is that the average number is mu times T.

So you can see that the firing rate, mu, is just the expected number of spikes divided by the width of the window. Does that make sense? That's pretty obvious. That's just the spike rate. And we're going to often use the variable r for that, for firing rate. OK. And here's what that looks like. You can see that if the firing rate is low enough or the window is short enough, such that the expected number of spikes is 1, you can see that, most of the time, you're going to get 0 or 1 spikes, and then occasionally 2, and very occasionally 3, and then almost never more than 4 or 5.

If the expected number of spikes is 4, you can see that the mode of that-- you can see that that distribution moves to the right. You have a higher probability of getting 3 or 4 spikes. But again, there's still a distribution. So even if the average number of

spikes is 4, you have quite a wide range of actual spike counts that you would get on any given trial. As the expected number of spikes gets bigger-- let's say, on average, 10-- does anyone know what this distribution starts turning into?

AUDIENCE: Gaussian.

MICHALE FEE: Gaussian, good. So what you can see is that you end up having a more symmetric distribution, where the peak is sitting at the expected number. In that case, the distribution becomes more symmetric, and in the limit of infinite expected number of spikes, becomes exactly a Gaussian distribution.

All right, there are two measures that we use to characterize how variable spike trains are. Le me just go through them real quick. And we'll describe-- and I'll describe to you what we expect those to look like for spike trains that have a Poisson distribution.

So the first thing we can look at is the variance in the spike count. So remember, for a Poisson process, the average number of spikes in the interval is mu times T. We can calculate the variance in that, which is basically just some measure of the width of this distribution here.

So the variance is just the number of counts on a given trial minus the expected number, squared. n minus average and, squared. And if you multiply that out, you get-- it's the average n squared minus the average squared. And it turns out, for the Poisson process, that that variance is also mu T. So the variance is-- the average spike count is mu T, and the variance in the spike count is also mu T.

So there is a quantity called the Fano factor, which is just defined as the variance of the spike count divided by the average spike count. And for a Poisson process, the Fano factor is 1. And what you find is that for neurons in cortex and other parts of the brain, the Fano factor actually can be quite close to one. It's usually between 1 and 1 and 1/2 or so.

So there's been a lot of discussion and interest in why it is that spike counts in the brain are actually so random, why do neurons behave in such a way that their spikes occur essentially randomly at some rate. So it's a interesting topic of current research interest.

OK, let me tell you about one other measure, called the interspike interval distribution. And basically the inner spike interval distribution is the distribution of times between spikes. And I'm just going to show you what that looks like in the Poisson process, and then briefly describe what that looks like for real neurons.

OK, so let's say we have a spike. OK, let's calculate the distribution of intervals between spikes. So let's say we have a spike at time T-sub-i. We're going to ask what is the probability that we have a spike some time, tau, later-- tau-sub-i later, within some little window, delta-T.

So let's calculate that. So tau-sub-i is initial spike interval between the i-plus-1 spike and the i-th spike. So the probability of having the next spike land in the interval between t of i plus 1 and t of i plus 1 plus delta t in this little window here is going to be what? It's going to be the probability of having no spike in this interval, times the probability of having a spike in that little interval. So what's the probability of having no spike in that interval, tau?

What distribution can we use to calculate that probability?

AUDIENCE: [INAUDIBLE]

MICHALE FEE: Yeah. So what is the Poisson distribution? It tells us the probability of having n spikes in an interval T-- capital T. So how would we use that to calculate the probability of having no spike in that interval?

AUDIENCE: [INAUDIBLE].

MICHALE FEE: Exactly. We just use the Poisson distribution, and plug n equals 0 into it. So go to that. There's the Poisson distribution. What does this look like if we set n equals 0? So what is mu T to the zero?

AUDIENCE: [INAUDIBLE]

MICHALE FEE: 1. What is 0 factorial?

AUDIENCE: 1.

MICHALE FEE: 1. And so the probability of having zero spikes in a window T is just e to the minus mu T. All right, so let's plug that in-- good. So the probability of having no spikes in

that interval is e to the minus mu T, or rT, if we're using r for rate now.

Now, what is the probability of having a spike in that little window right there? Any thoughts?

AUDIENCE: [INAUDIBLE]

MICHALE FEE: What's that?

AUDIENCE: [INAUDIBLE]

MICHALE FEE: Yeah. You could do that. But we sort of derive the Poisson process by using the answer to this question already.

AUDIENCE: Oh, r delta.

MICHALE FEE: r delta-T. Good. OK, so the probability of having a spike in that little window is just r delta-T. OK. Yes.

AUDIENCE: Stupid question-- I just missed the transition between u and r.

MICHALE FEE: Yeah, I just changed the name of the variable. If you look in the statistics literature, they most often use mu. But when we're talking about prime rates of neurons, it's more convenient to use R. And so they're just the same.

AUDIENCE: [INAUDIBLE]

MICHALE FEE: Yes.

AUDIENCE: Are we talking the probability of having the next five commands [INAUDIBLE] given that there was no spike at the first interval?

MICHALE FEE: Well, so remember, the probability of having a spike in any interval in a process is completely independent of what happens at any other time.

AUDIENCE: So why are we calculating-- why did we do that [INAUDIBLE]?

MICHALE FEE: OK, because in order for this to be the interval between one spike and the next spike, we needed to have zero spikes in the intervening interval.

AUDIENCE: Oh, OK, so [INAUDIBLE]. OK.

MICHALE FEE: Because if there had been another spike in here somewhere, this would not be our inter-spike interval. The inter-spike interval would be between here and wherever that spike occurred. And we'd just be back to calculating what's the probability of having no spike between there and there.

OK, good. So that's the probability of having no spike from here to here, and having a spike in the next little delta-T. We can now calculate what's called the probability density. It's the probability per unit time of having inter-spike intervals of that duration. And to do that, we just calculate the probability divided by delta-T. And what you find is that the probability density of inter-spike intervals is just re to the minus r tau, where tau is the spike interval. And so this is what that looks like for a Poisson process.

OK, so you can see that the highest probability is having very short intervals. And you have exponentially lower probabilities of having longer and longer intervals. Does that make sense? So it turns out that that's actually a lot like what interspike intervals of real neurons looks like. They very often have this exponential tail.

Now, what is completely unrealistic about this interspike interval distribution? What is it that can't be true about this? [INAUDIBLE].

AUDIENCE: Intervals, like, which are huge.

MICHALE FEE: Well, so it's actually-- the bigger problem is not on this end of the distribution.

AUDIENCE: [INAUDIBLE]

MICHALE FEE: Yeah. What happens here that's wrong?

AUDIENCE: [INAUDIBLE]

MICHALE FEE: Exactly. So what happens immediately after a neuron spikes?

AUDIENCE: You can't have a spike.

MICHALE FEE: You can't have another spike right away. Why is that?

AUDIENCE: [INAUDIBLE]

MICHALE FEE: Because of the refractory period, which comes from--

AUDIENCE: [INAUDIBLE]

MICHALE FEE: Hyperpolarization is one of them. Once you have the spike, the neuron is actually briefly hyperpolarized. You could imagine trying to re-polarize it very quickly, but then something else is a problem.

AUDIENCE: [INAUDIBLE]

MICHALE FEE: Sodium channel inactivation. So even if you were to repolarize the neuron very quickly, it still would have a hard time making a spike because of sodium channel inactivation. So what does this actually look like for real neuron? What do you imagine it looks like?

AUDIENCE: [INAUDIBLE]

MICHALE FEE: Right, so immediately after a spike, there is zero probability of having another spike. So this starts at zero, climbs up here, and then decays exponentially. OK, so that's what most inter-spike intervals for real neurons actually looks like.

So this is probability density. This is tau. And this is the refractory period. In fact, when you record from neurons with an electrode in the brain, you get lots of spikes. One of the first things you should do when you set a threshold and find those spike times is compute by interval distribution. Because if you're recording from a single neuron, that interspike interval distribution will have this refractory period. If you're recording from-- it's quite easy to get multiple spikes on the end of an electrode.

And what happens if you have multiple spikes, you can think they're coming from one neuron, but in fact they're coming from two neurons. And if you compute the interspike interval distribution, it won't have this dip here. So it's a really important tool to use to test whether your signal is clean. You'd be amazed at how few people actually do that.

All right, I just want to introduce you to one important term. And that's called homogeneous versus inhomogeneous in the context of Poisson process. What that means is that a homogeneous Poisson process has a constant rate, mu. Most

neurons don't do that. Because in most neurons, there's information carried in the fluctuation of the firing rate. And so most neurons have what's called behave more like an inhomogeneous Poisson process, where the rate is actually fluctuating in time. And the example we were looking at before shows you what an inhomogenous Poisson process would look like.

So let's see what's next. All right, so let's change topics to talk about convolution, cross-correlation, and auto-correlation functions. All right, so we've been using the notion of a convolution where we have a kernel that we multiply by a signal. We multiply that, and integrate over time, and then we slide that kernel across the signal, and ask, where does the signal have structure that looks like the kernel. And that gives you an output y of T.

So we've used that to model the membrane potential of [INAUDIBLE] synaptic input. So in that case, we had spikes coming from a presynaptic neuron that generate a response in the postsynaptic neuron. And now we can take-- the output of that, the response in the postsynaptic neuron as a convolution of that exponential, with an input spike train. We've used it to model the response of neurons to a time-dependent stimulus.

And you also used it-- I've described how you can use that to implement a low-pass filter or a high-pass filter. And we talked about doing that for extracting either low-frequency signals, to get the local field potential out of neurons, or doing a high-pass filter to get rid of the low-frequency signals so that you can see the spikes.

OK, but most generally, convolution is-- you should think about it as allowing you to model how a system responds to an input where the response of the system is controlled by a linear filter. The system is sensitive to particular patterns in the input. Those patterns can be detected by a filter. So you apply that filter to the input, and you estimate how the system responds. And it's very broadly useful in engineering and biology and neuroscience to use convolutions to model how a system responds to its input.

All right, so there is a different kind of function, called a cross-correlation function, that looks like this. And it looks very similar to a convolution, but it's used very differently. Now, you might think, OK, there's just a sign change. Here I have a T

minus tau, and here I have a T plus tau. Is that the only difference between those? It's not. Because here I'm integrating over a d tau, and here I'm integrating over dT. Here I'm getting the response as a function of time.

Here, what I'm doing is I'm kind of extracting the kernel. What I'm getting out of this is a kernel, tau, as a function of tau. OK, so let me walk through how that works. So in this case, we have two signals, x and y. And what we're doing is we're taking those two signals and we're multiplying them by each other, x times y. And what we're doing is we're multiplying the signals and then integrating over the product. And then we shift one of those signals by a little amount, tau. And we repeat that process.

So let's see what that looks like. So what do you see here? You see two different signals, x and [AUDIO OUT] I'm sorry, x and y. They look kind of similar. You can see this one has a bump here, and a couple bumps there, and a bump there. And you see something pretty similar to that here, right? Here's three big bumps. Here's three big bumps. So those signals are actually quite similar to each other, but they're not exactly the same. Right you can see that these fast little fluctuations are different on those two signals. Now, what happens when we take this signal-- and you can see that there's this offset here.

So what happens if we take x times y, we multiply them together, and we integrate the result? Then we shift y a little bit, by an amount, tau, and we multiply those two signals together, and we integrate. And we keep doing that. And now we're going to plot this k as a function of that little shift, tau, that we put in there.

And here's what that looks like. So k is the cross-correlation-- sometimes called the lag cross-correlation-- of x and y. So that little circle is the symbol for lag cross-correlation. And you can see that, at a particular lag, like right here, the positive fluctuations in this signal are going to line up with the positive fluctuations in that signal. The negative fluctuations in this signal are going to line up with the negative fluctuations in that signal. And when you multiply them together, you're going to get the maximum positive contribution. Those signals are going to be maximally overlapped at a particular shift, tau.

And when you put that function, K of tau, you're going to have a peak at that lag

that corresponds to where those signals are maximally overlapped. So a lag cross-correlation function is really useful for finding, let's say, the time at which two signals-- like if one signal is like a copy of the other one, but it's shifted in time, a lag cross-correlation function is really useful for finding that the lag between those two functions.

There's another context in which this lag cross-correlation function is really useful. So when we did the spike-triggered average, when we took spikes from a neuron, and we-- at the time of those spikes, we extracted what the input was to the neuron, and we averaged that [AUDIO OUT]. What we were really doing was we were doing a cross-correlation between the spike train of a neuron and the stimulus that drove that neuron, the stimulus that was input to that neuron. And that cross-correlation was just the kernel that described how you get from that input to the neuron to the response of the neuron.

So spike-triggered average is actually sometimes called reverse correlation. And the reverse comes from the fact that you have to actually flip this over to get the kernel. So let's not worry about that. But it's sometimes referred to as the correlation between the spike train and the stimulus input. Yes.

AUDIENCE: [INAUDIBLE]

MICHALE FEE: So there's no convolution here. We're doing the lag cross-correlation, OK. We're just taking those two signals, and shifting one of them, multiplying, and integrating. Sorry. Ask your question. I just wanted to clarify, there's no convolution being done here.

AUDIENCE: So [INAUDIBLE] is that y [INAUDIBLE]

MICHALE FEE: Ah, OK, great. That's actually one of the things that's easiest to get mixed up about when you do lag cross-correlation. You have to be careful about which side this peak is on. And I personally can never keep it straight. So what I do is just make two test functions, and stick them in. And make sure that if I have one of my functions-- so in this case, here are two test functions. You can see that x is-- sorry, y is before x. And so this peak here corresponds to y happening before x. So you just have to make sure you know what the sign is. And I recommend doing that with just two

functions that you know. It's easy to get it backwards. Yes.

AUDIENCE: [INAUDIBLE] part of y equals [INAUDIBLE].

MICHALE FEE: What do you mean, first part?

AUDIENCE: The very first [INAUDIBLE].

MICHALE FEE: Oh, these?

AUDIENCE: Yes.

MICHALE FEE: I'm just taking copies of this, and placing them on top of each other, and shifting them. So you should ignore this little part right here.

AUDIENCE: [INAUDIBLE]

MICHALE FEE: Oh, yeah. So there are Matlab functions to do this cross-correlation. And it handles all of that stuff for you. Yes.

AUDIENCE: [INAUDIBLE]

MICHALE FEE: Yes.

AUDIENCE: Like, is the x-axis tau?

MICHALE FEE: The x-axis is tau. I should have labeled it on there.

AUDIENCE: I just want to clarify whether they meet. So how it's kind of flat, and then this idea of the flat area that, when you multiply them and they're slightly off, this is generally not zero. But you know, they cancel each other out.

MICHALE FEE: So you can see that if these things are shifted-- if they're kind of noisy, and they're shifted with respect to each other, you can see that the positive parts here might line up with the negative parts there. And the negative parts here line up with the positive parts there. And they're shifted randomly at any other time with respect to each other. On average, positive times negative, you're going to have some parts where it's positive times positive, and other parts where it's positive times negative. And it's all going to wash out and give you zero. Because those signals are uncorrelated with each other at different time lags.

AUDIENCE: So only when the perfectly overlap and kind of exaggerate each other is when they'll be flat.

MICHALE FEE: Exactly. It's only when they overlap that all of the positive parts here will line up with the positive parts here, and the negative parts here will line up with the negative parts.

AUDIENCE: And since you're taking the integral together, it just makes a very large positive.

MICHALE FEE: Exactly. All of those positive times positive add up to the negative times negative, which is positive. And all of that adds up to give you that positive peak right there. But when they're shifted, all of those magical alignments of positive with positive and negative with negative go away. And it just becomes, on average, just random. And random things have zero correlation with each other.

So this is actually a really powerful way to discover the relation between two different signals. And in fact, it gives you a kernel that you can actually use to predict one signal from another signal. Now, if you were to take x and convolve it with k, you would get an estimate of y. We'll talk more about that later. Pretty cool, right?

So they're mathematically very similar. They look similar. But they're used in different ways. So the way we think about convolution is-- what it's used for is to take an input signal x, and convolve it with a kernel, k, to get an output signal, y. And we usually think of x as being very long vectors, long signals. K here, kappa, is a kernel. It's just a short little thing in time. And you're going to convolve it with a long signal to get another long signal. Does that makes sense?

In cross-correlation, we have two signals, x and y. x and y are the inputs. And we cross-correlate it to extract a short signal that captures the temporal relation between x and y. All right, so x and y are long signals. K is a short vector. And in this case, we're convolving a long signal with a short kernel to get another long signal, which is the response of the system. In this case, we take, let's say, the input to a system and the output of this system, and doing a cross-correlation to extract the kernel.

All right, any questions about that? They're both super-powerful methods. Yes, [INAUDIBLE].

AUDIENCE: How does the cross-correlation-- mathematically, how does it give a short vector [INAUDIBLE]?

MICHALE FEE: You just do this for a small number of taus. Because usually signals that have some relation between them, that relation has a short time extent. This signal here, in a real physical system, doesn't depend on what x was doing a month ago, or maybe a few seconds ago. These signals might be 10 or 20 seconds long, or an hour long, but this signal, y, doesn't depend on what x was doing an hour ago. It only depends on what x was doing maybe a few tens of milliseconds or a second ago. So that's why x and y can be long signals. K is a short vector that represents the kernel.

Any more questions about that? OK, these are very powerful methods. We use them all the time. I talked about the relation to the spike-triggered average.

The autocorrelation is nothing more than a cross-correlation of a signal with itself. So let's see how that's useful. So an autocorrelation is a good way to examine a temporal structure within a signal. So if we take a signal, x, and we calculate the cross-correlation of that signal with itself, here's what that looks like. So let's say we have a signal that looks like this. And this signal kind of has slowish fluctuations that are, let's say, on a 50 or 100 millisecond timescale.

If we take that signal and we multiply it by itself with zero time lag, what do you think that will look like? So positive lines up with positive, negative lines up with negative. That thing should do what? It should be a maximum, right? Autocorrelations are always a maximum at zero lag.

And then what we do is we're just going to take that signal and shift it a little bit. And we'll shift it a little bit more, do that product and integral, shift that product and integral. Now what's going to happen as we shift one of those signals sideways? And then multiply and integrate. Sammy, you got this one.

AUDIENCE: Oh, at first, [INAUDIBLE] zero [INAUDIBLE] point where the plus and minuses cancel out. [INAUDIBLE] this. If you [INAUDIBLE] maybe [INAUDIBLE] where the pluses overlap the second time.

MICHALE FEE: Yeah, so if you shift it enough, it's possible that it might overlap again somewhere else. What kind of signal would that happen for?

AUDIENCE: Like, cyclical.

MICHALE FEE: Yeah, a periodic signal. Exactly. So an autocorrelation first of all has a peak at zero lag. That peak drops off when these fluctuations here. So the positive lines up with positive, negative lines up with negative. As you shift one of them, those positive and negatives no longer overlap with each other, and you start getting positives lining up with negatives. And so the autocorrelation drops off. And then it can actually go back and indicate-- it can go back up if there's periodic structure.

So let's look at what this one looks like. So in this case, it kind of looked like there might be periodic structure, but there wasn't, because this was just low-pass-filtered noise. And you can see that if you compute the-- there's that Matlab function, xcorr. So you use it to calculate the autocorrelation. You can see that that autocorrelation is peaked at zero lag, drops off, and the lag at which it drops off depends on how smoothly varying that function is. If the function varies-- if it's changing very slowly, the autocorrelation is going to be very wide. Because you have to shift it a lot in order to get the positive peaks and the negative peaks misaligned from each other.

Now, if you have a fast signal like this, with fast fluctuations, and you do the autocorrelation, what's going to happen? So first of all, at zero lag, it's going to be peak. And then what's going to happen? It's going to drop off more quickly. So that's exactly what happens.

So here's the autocorrelation of this slow function. Here's the autocorrelation of this fast function. And you can see both of these are just noise. But I've smoothed this one. I've low-pass-filtered this with kind of a 50-millisecond-wide kernel to give very slowly-varying structure. This one I smoothed with a very narrow kernel to leave fast fluctuations. And you can see that the autocorrelation shows that the width of the smoothing of this signal is very narrow. And the width of the smoothing for this signal was broad. In fact what you can see is that this signal looks like noise that's been convolved with this, and this signal looks like noise that's been convolved with that.

And here is actually a demonstration of what Sammy was just talking about. If you take a look at this signal, so the autocorrelation can be a very powerful method. It's actually not that powerful. It's a method for extracting periodic structure. And we're going to turn now, very shortly, to a method that really is very powerful for extracting periodic structure.

But I just want to show you how this method can be used. And so if you look at that signal right there, it looks like noise. But there's actually structure embedded in there that we can see if we do the autocorrelation of this signal. And here's what that looks like. So again, autocorrelation has peaked. The peak is very narrow, because that's a very noisy-looking signal. But you can see that, buried under there, is this periodic fluctuation. What that says is that if I take a copy of that signal and shift it with respect to itself every 100 milliseconds, something in there starts lining up again. And that's why you have these little peaks in the autocorrelation function.

And what do you think that is buried in there? The sine wave.

So this data is random, with a normal distribution, plus 0.1 times cosine that gives you a 10 hertz wiggle.

So you can't see that in the data. But if you do the autocorrelation, all of a sudden you can see that it's buried in there. All right, so cross-correlation is a way to extract the temporal relation between different signals. Autocorrelation is a way to use the same method essentially to extract the temporal relation between a signal and itself at different times. And that method, it's actually quite commonly used to extract structure and spike trains. But there are much more powerful methods for extracting periodic structure. And that's what we're going to start talking about now.

OK, any questions? Let's start on the topic of spectral analysis, which is the right way to pull out the periodic structure of signals. Here's that the spectrogram that I was actually trying to show you last time.

So what is a spectrogram? So if we have a sound-- we record a sound with a microphone, microphones pick up pressure fluctuations. So in this case, this is a bird singing. The vocal cords are vibrating with air flowing through them that produce very large pressure fluctuations in the vocal tract. And those are transmitted out through the beak, into the air. And that produces pressure fluctuations that

propagate through the air at about a foot per millisecond. They reach your ear, and they vibrate your eardrum.

And if you have a microphone there, you can actually record those pressure fluctuations. And if you look at it, it just looks like fast wiggles in the pressure. But somehow your ear is able to magically transform that into this neural representation of what that sound is. And what your ear is actually doing is it's doing a spectral analysis of the sound.

And then your brain is doing a bunch of processing that helps you identify what that thing is that's making that sound. So this is a spectral analysis of this bit of birdsong, canary song. And what it is, it does very much what your eardrum does-- or what your cochlea does. Sorry. It calculates how much power there is at different frequencies of the sound and at different times.

So what this says is that there's a lot of power in this sound at 5 kilohertz at this time, but at this time, there's a lot of power at 3 kilohertz or a 2 kilohertz and so on. So it's a graphical way of describing what the sound is. And here's what that looks like.

[DESCENDING WHISTLING BIRDSONG]

[FULL-THROATED CHIRP]

[RAPID CHIRPS]

[SIREN-LIKE CHIRPS]

[STUDENTS LAUGH]

So you can see visually what's happening. Even though if you were to look at those patterns on an oscilloscope or printed out on the computer, it would just be a bunch of wiggles. You wouldn't be able to see any of that.

Here's another example. This is a baby bird babbling. And here's an example of what those signals actually look like-- just a bunch of wiggles. It's almost completely uninterpretable. But again, your brain does this spectral analysis. And here I'm

showing a spectrogram. Again, this is frequency versus time. It's much more cluttered visually. And it's a little bit harder to interpret. But your brain actually does a pretty good job figuring out--

[RANDOM SQUEAKY CHIRPING]

--what's going on there.

We can also do spectral analysis of neural signals. So one of the really cool things that happens in the cortex is that neural circuits produce oscillations. And they produce oscillations at different frequencies at different times. If you close your eyes-- everybody just close your eyes for a second. As soon as you close your eyes, the back of your cortex starts generating a big 10 hertz oscillation. It's wild. You just close your eyes, and it starts going--

[MAKES OSCILLATING SOUND WITH MOUTH]

--at 10 hertz.

This is a rhythm that's produced by the hippocampus. So whenever you start walking, your hippocampus starts generating a 10 hertz rhythm. When you stop and you're thinking about something or eating something, it stops. As soon as you start walking--

[MAKES OSCILLATING SOUND WITH MOUTH]

--10 hertz rhythm. And you can see that rhythm often in neural signals. Here you can see this 10 Hertz rhythm. It has a period of about 100 milliseconds. But on top of that, there are much faster rhythms. It's not always so obvious in the brain what the rhythms are. And you need spectral analysis techniques to help pull out that structure.

You can see here, here is the frequency as a function of time. I haven't labeled the axis here. But this bright band right here corresponds to this 10 hertz oscillation.

So we're going to take up a little bit of a detour into how to actually carry out state-of-the-art spectral analysis like this to allow you to detect very subtle, small signals in neural signals, or sound signals, or any kind of signal that you're interested in

studying. Jasmine.

AUDIENCE: [INAUDIBLE]

MICHALE FEE: Yeah, OK, so this is called a color map. I didn't put that on the side here. But basically dark here means no power. And light blue to green is more power. Yellow to red is even more. Same here. So red is most power.

AUDIENCE: [INAUDIBLE]

MICHALE FEE: Yes. So it's how much energy there is at different frequencies in the signal as a function of time.

So you're going to know how to do this. Don't worry. you're going to be world experts at how to do this right. Yes.

AUDIENCE: [INAUDIBLE]

MICHALE FEE: Yes.

AUDIENCE: This is very similar to [INAUDIBLE].

MICHALE FEE: Yes. I'm sorry. I should have been clear-- this is recording from an electrode in the hippocampus. And these oscillations here are the local field potentials.

AUDIENCE: [INAUDIBLE]

MICHALE FEE: Yes. That's exactly right. Thank you. I should have been more clear about that. OK, all right, so in order to understand how to really do this, you have to understand Fourier decomposition. And once you understand that, the rest is pretty easy. So I'm going to take you very slowly-- and it may feel a little grueling at first. But if you understand this, the rest is simple. So let's just get started.

All right, so Fourier series-- it's a way of decomposing periodic signals by applying filters to them. We're going to take a signal that's a function of time, and we're going to make different receptive fields. We're going to make neurons that have-- sensitive to different frequencies. And we're going to apply those different filters to extract what different frequencies there are in that signal.

So we're going to take this periodic signal as a function of time. It's a square wave. It has a period capital T. And what we're going to do is we're going to approximate that square wave as a sum of a bunch of different sine waves. So the first thing we can do is approximate it as a cosine. You can see the square wave has a peak there. It has a valley there. So the sine wave approximation is going to have a peak there and a valley there. We're going to approximate it as a cosine wave of the same period and the same amplitude.

Now, we might say, OK, that's not a very good approximation. What could we add to it to make a better approximation?

AUDIENCE: [INAUDIBLE]

MICHALE FEE: Another cosine wave, which is what we're going to do in just a second. Because apparently there's more I wanted to tell you about this. So we're going to approximate this as a cosine wave that has a coefficient in front of it. We have to approximate this is a cosine that has some amplitude a1, and it has some frequency, f0. So a cosine wave with a frequency f0 is this-- cosine 2 pi f0 T, where f0 is 1 over the period. And you can also write-- so f0 is the oscillation frequency. It's cycles per second.

There is an important quantity called the angular frequency, which is just 2 pi times f0, or 2 pi over T. And the reason is because if we write this as cosine omega-0 T, then this thing just gives us 2 pi change in the phase over this interval T, which means that the cosine comes back to where it was.

So if we want to make a better approximation, we can add more sine waves or cosine waves. And it turns out that we can approximate any periodic signal by adding more cosine waves that are multiples of this frequency, omega-0. Why is that? Why can we get away with just adding more cosines that are integer multiples of this frequency, omega-0? Any idea?

Why is it not important to consider 1.5 times omega-0, or 2.7?

AUDIENCE: Does it have something to do with [INAUDIBLE]?

MICHALE FEE: It's close, but too complicated.

AUDIENCE: [INAUDIBLE] sum just over the period of their natural wave, right? So if you add non-integer [INAUDIBLE], you lose that [INAUDIBLE].

MICHALE FEE: Right. This signal that we're trying to model is periodic, with frequency omega-0. And any integer multiple of omega-0 is also periodic at frequency omega-0.

So notice that this signal that's cosine 2 omega-0 is still periodic over this interval, t. Does that make sense? So any integer multiple cosine integer omega 0 is also periodic with period T. OK? And those are the only frequencies that are periodic with period T.

So we can restrict ourselves to including only frequencies that are integer multiples of omega-0, because those are the only frequencies that are also periodic with period T.

We can write down-- we can approximate this square wave as a sum of cosine of different frequencies. And the frequencies that we consider are just the integer multiples of omega-0. Any questions about that? That's almost like the crux of it. There's just some more math to do.

So here's why this works. So here's the square wave we're trying to approximate. We can approximate that square wave as a cosine. And then there's the approximation. Now, if we add a component that's some constant times cosine 3 omega-0, you can see that those two peaks there kind of square out those round edges of that cosine. And you can see it starts looking a little bit more square. And if you add constant times cosine 5 omega-T, it gets even more square. And you keep adding more of these things until it almost looks just like a square wave.

Here's another function. We're going to approximate a bunch of pulses that have a period of 1 by adding up a bunch of cosines. So here's the first approximation-- cosine, omega-0. So there's your first approximation to this train of pulses. Now we add to that a constant times cosine 2 omega-T. See that peak gets a little sharper. Add a constant times cosine 3 omega-T. And you keep adding more and more terms, and it gets sharper and sharper, and looks more like a bunch of pulses.

Why is that? It's because all of those different cosines add up consecutively right here at 0. And so those things all add up. And this peak stays, because the peak of

those cosines is always at 0. Over here, though, right next door, all of those, you can see that that cosine is canceling that one. It's canceling that one. Those two are canceling. You can see all these peaks here are canceling each other. And so that goes to 0 in there.

Now of course all these cosines are periodic with a period T. So if you go one T over, all of those peaks add up again and interfere constructively. So that's it. It's basically a way of taking any periodic signal and figuring out a bunch of cosines such that the parts you want to keep add up constructively and the parts that aren't there in your signal add up destructively. And there's very simple sort of mathematical tools-- basically it's a correlation-- to extract the coefficients that go in front of each one of these cosines.

And then one more thing-- we use cosines to model or to approximate functions that are symmetric around the original. Because the cosine function is symmetric. Other functions are anti-symmetric. They'll look more like a sine wave. They'll be negative here, and positive there. And we use sines to model those.

And then the one last trick we need is we can model arbitrary functions by combining sines and cosines into complex exponentials. And we'll talk about that next time. And once we do that, then you can basically model not just periodic signals, but arbitrary signals. And then you're all set to analyze any kind of periodic signal in arbitrary signals. So it's a powerful way of extracting periodic structure from any signal. So we'll continue that next time.