[SQUEAKING]

[RUSTLING]

[CLICKING]

PROFESSOR: So today, for the first half of the lecture, we're going to have another guest lecturer, Gaurav Arya, who's actually an undergraduate at MIT. But he's done research projects with me, and also research projects with Alan Edelman's group, and he's done some really interesting recent work on differentiating random functions, like you program, you flip a coin, and 50% it gives you sine, 50% chance it gives you cosine. So what does it mean to differentiate such a thing, and how do you do it automatically?

And so I thought that would be a-- we thought that would be a nice topic for half a lecture. So he's in Hong Kong right now, but he's going to give a guest lecture. So I think we can get started, Gaurav.

GAURAV ARYA: OK. Yeah, so I'm-- can you hear me all right?

PROFESSOR: Can anyone hear him? Good. Good, yeah.

GAURAV ARYA: OK. Yeah, yeah. I see myself.

PROFESSOR: Yeah, yeah. So I'll try and repeat any questions and things like that.

GAURAV ARYA: Yeah. So I'm a junior at MIT. So yeah, as Professor Johnson said. I did a year with Professor Johnson previously. And recently, I've been doing a year in the Julia Lab, which is led by Professor Edelman.

And today, I'm going to be talking about derivatives of random functions. So I'm going to try and do this. Push right-- oops. That's not what I wanted to do.

Yeah. So I wanted to start by just giving a bit of motivation. So I'm going to just-- before I go into the random functions, I'm going to write this function, which I think you all probably know how to differentiate better than I do at this point. So here, A is a matrix, and this is a matrix-accepting function. Can you read this all fine?

PROFESSOR: Yep. We can see it.

GAURAV ARYA: Yeah. So this is a function with matrix input and matrix output. And the reason I want to go through this again is just to set up an analogy with how I-- with differentiating random functions, what are the broad steps that 1 takes in-- when you're faced with a function like this.

The essential complication is that your input and output spaces are much more complicated, and they're not real-to-reals, like in single variable calculus. So when you want to differentiate a function like this, it becomes a lot harder to just do it super mechanically like you might do for single variable calculus. And it becomes a lot more appealing to go back to the fundamental questions that you ask when taking a derivative.

So the first might be something like-- ultimately, derivatives are about when you perturb the input slightly, how does the output perturb? A map from input wiggles, the output wiggles. So the first question you might ask is just, if I perturb the input, how does the output change?

So perturb the input. How does the output change? What's the sensitivity of the output with respect to the input? So for the case of this function, what you've been-- what you've done in the class is you may have written something like this. You might have used this notation df, which is the differential of f.

PROFESSOR: Yep. That's the notation. Yeah.

GAURAV ARYA: Yeah. Yep. Sorry?

PROFESSOR: That's the notation, yes.

GAURAV ARYA: Yeah, OK. And you would have replaced A with A plus dA. So the A is your wiggle and the input, this is the differential. So we subtract off the original A squared.

You can expand this out. So the A squares cancel, so you have-- you've done this, I'm sure, quite a few times. Then we have this dA squared term here. And this--

now, I know that the next step you're probably eager to do is to cancel out the dA squared. But just this on its own is the answer to question 1. We're not worried about neglecting the higher order terms or forming the derivative, but just thinking about how do we write down the change in the output. And this is your function. So this is your answer to question 1.

And then the second thing is you have this differential, and you want to figure out what terms can you neglect here. So what terms do I keep, what terms do I neglect? Because, really, our goal at this point is to form this derivative, which the idea of the derivative is to capture the sensitivity of f with respect to A to the first order. The most important effects to the first order, and we have to drop the rest.

So the second question is what terms to neglect, like that. Yeah. And here, you can just cross off the dA squared. That's a higher order term. So you can justify it that way.

And your derivative operator becomes this map from dA to what's left here. Right? The dA times A plus A times dA. So I just wanted to separate out these two steps, because both of them are kind of non-trivial in the case of random functions.

So the first is just thinking about this map from input perturbations to output perturbations. And the second is, OK, great. I understand that. But now what terms should I neglect to form this derivative? Does that make sense? Are there any questions?

PROFESSOR: Any questions? Nope. People are shaking their heads.

GAURAV ARYA: Great.

PROFESSOR: They've seen this 100 times.

GAURAV ARYA: So that is a matrix-to-matrix function. But the idea there was just thinking about a function with a more complicated input space and a more complicated output space. So now we're going to shift our attention to another case where we were trying to generalize the notion of the derivative to more complicated spaces. And this time, we're considering functions which are random.

So more concretely, I'm going to write a random function with a capital letter X, which already might be a bit suggestive, because capital letters are often used for denoting random variables. And indeed, this is a random variable value function. So as a map, we might write that it takes in some input P, and it spits out X of P, where X of P is a random variable. And I'm going to keep-- the output space is pretty complicated, so I'm going to keep the output space fairly simple.

The input, let's assume, is just a single real number. So P belongs to the reals. So yeah. So now we want to ask the same question. How--

So for our matrix function, we started with this map from to A to A squared, and we got as our answer for the derivative another map, map from dA to dA times A plus A times dA. And now we want to ask, what is a useful notion of derivative?

First of all, what is our notion of differential? How does the perturbation of the input affect-- if every time I run this function I get a different value if it's a random function, what is a useful notion of sensitivity of the output, and so forth? So we want to do the same process on this random function.

PROFESSOR: So does everyone know what he means by a "random variable." Like I said, suppose you flip a coin, and if it's heads, you return 1. And if it's tails, you return 0. All right? That would be a discrete random variable.

GAURAV ARYA: Yeah. Why don't I actually show some examples? So let me first write them down.

PROFESSOR: Yes.

GAURAV ARYA: So we could have, let's say, a random variable-- and one crucial thing is-- so this is a random variable value function. So it's actually a whole family of random variables indexed by parameter P. So let's say one example you could have is X of P could be following the Bernoulli distribution with probability P. This is your example of probability P of heads and otherwise tails. Then maybe another example--

PROFESSOR: So that yeah. With probability P, you return 1, and with probability--

GAURAV ARYA: Yeah.

PROFESSOR: --1 minus P returns 0, for example.

GAURAV ARYA: Exactly. Yeah. And maybe another example. So I'm going to write an exponential distribution here, and I'm not hoping to really say what it is.

But the point is, this one's a bit different in nature from the Bernoulli, because it's like a continuous distribution. And I'll show you how examples from that distribution look as well. So let me just try and share my screen. Can you see--

PROFESSOR: Yep.

GAURAV ARYA: You can see well?

PROFESSOR: We can see your screen, yeah.

GAURAV ARYA: OK. So yeah. So let's consider the Bernoulli example. So X of P-- so really, what I'm writing here is a program that samples from X of P.

So I'll call it sampleX(p). And in this case, this is basically a sampling from a Bernoulli distribution. So this thing inside of the rand() is the distribution. But we're getting samples from it, so that's why there's this rand in front.

OK, so now I can run this program with X of 0.6 and sampling from X of 0.6. In this case, it's a 1 or 0. In this case, it's using a Boolean to represent the output. But the point is that you're going to get true 60% of the time and false 40% of the time. So you can verify this by taking a bunch of samples.

PROFESSOR: In Julia, true is equivalent to the number 1, and false is equivalent to the number 0.

GAURAV ARYA: Yeah. And then let's just stare at the other example of an exponential distribution. So the Bernoulli is what's called a discrete random variable. And now for an exponential distribution, maybe I'll make P 100.

And now it's doing some continuous random thing. But the point is you get a different output every time. Now, the question is--

PROFESSOR: Do people know what the exponential--

GAURAV ARYA: --differentiation. So someone codes up a function--

PROFESSOR: One sec. One second.

GAURAV ARYA: Yeah.

PROFESSOR: Do people know what the exponential distribution means? It means the probability-- you're giving a non-negative real number. It's returning continuously out. But the probability decays exponentially as it gets larger.

And so P here is just the scale of that exponential. So your-- the real number is in the order of-- sampleX(100) is real numbers in the order of 100. It can be larger. It can be smaller. But as it gets bigger than 100, it gets exponentially less likely, and then follows an exponential curve.

GAURAV ARYA: Right. Yeah.

PROFESSOR: But it's continuous numbers that are coming out.

GAURAV ARYA: Yeah, so I should have said that. So here, I'm parameterizing exponential distribution by its scale. Yeah.

So another question is, what is a useful notion of derivative for programs like this? Someone codes up a program that samples from an exponential distribution. What's a useful notion of derivatives?

So there's two questions here. There's how do we find this useful notion of derivative, but there's also the why. Why would we-- for the functions you considered so far, matrix value functions, the why is pretty clear, in that gradients are useful for optimizing functions and so forth.

And if you have a totally deterministic function, then it's very clear what it means to optimize that function. You have some objective and you're trying to make it as small as possible, and that's why you want to take gradients. But for these random functions, maybe it's not immediately obvious why we want any notion of derivative for these programs.

So why do we want to differentiate these? And maybe this is a question I can ask to the class. Why might we want to differentiate these sorts of programs?

PROFESSOR: Does anyone have any notion of why you would want to differentiate whatever deri-- we haven't defined what it means, but why would we want the sensitivity, in some sense, of that program with respect to some parameter, even though the outputs are random? Yeah. Someone raised-- yeah?

AUDIENCE: Maybe maximize the likelihood of something occurring?

PROFESSOR: Maybe to maximize the likelihood of something occurring, for example. That's--

GAURAV ARYA: OK. Yeah, yeah, yeah.

PROFESSOR: Another one? Yeah?

AUDIENCE: In physics, there are lots of systems with disorder that are random, in some sense. And if you'd like to measure properties of that system, then you need to take the randomness into account.

PROFESSOR: Yeah. So if you have a physical system with randomness, and you want to measure the sensitivity of something, some statistical quantity to--

GAURAV ARYA: Exactly, yeah.

PROFESSOR: --some physical parameter. Right.

GAURAV ARYA: Yeah, yeah. So both those answers are, yeah, totally correct. So basically, the idea is maybe you're normally interested in some statistical quantities of your system.

So X is your random variable. But ultimately, it might be computing some stochastic estimate which, on average, is going to give you something you want to minimize, for example. So I misspelled that.

So the physical quantity basically mean things that are constructed using averages. So they're based on expectations. And you may want to get these expectations somewhere you want them, and you want to find a P that maybe minimizes some expectation, and so forth.

So broadly, this is the why. And there's one more other point today, which might-- which is that, sure, you have this system which is, in principle, stochastic. But you're interested in maybe its average value. So why not just write a program which directly computes that average value, and is therefore deterministic?

If you have X of P, then the program you might be interested in is actually the expectation of X of P's average value. And this is a deterministic function of P. So why not just code that up and differentiate that? And the answer why we don't do that sometimes--

PROFESSOR: So quick question.

GAURAV ARYA: Yeah?

PROFESSOR: So quick question. So for the Bernoulli function that returned 1 with probability P and 0 with probability 1 minus P, what's the expectation value of that? It's just P. Right? So yeah. Yeah.

GAURAV ARYA: Yeah.

PROFESSOR: And P is an easy function to differentiate with respect to P. Right?

GAURAV ARYA: Yeah, yeah. So that's right. But you might have, for example, a very simple-- a random walk. In maybe five lines of code with a for loop, you can write a random walk which walks left-to-right.

Maybe it changes the probability of left-to-right depending on where you are. All that you can do in five lines of code. And now it becomes a lot easier to sample from this process than to analytically compute its average.

To compute its average, you might have to look at the transition matrix or something like that. But it's super easy to sample from the process. So that's the point. It's often a lot easier to sample.

And therefore, these samples are often called unbiased estimates, because on average, they're giving you the right thing than to compute analytically. Exactly compute. Yeah.

So then maybe I'll also give some more concrete examples. So one example is in deep learning, in ML, there is this very popular architecture called the variational autoencoder, or VAE.

And this architecture has randomness baked into its model. It's not just randomness from selecting a subset of examples to train on, because it's intrinsically random. So the loss function of this VAE would again be defined strictly speaking as an expectation.

But computationally, the model gives you the ability to sample from this X of P. That's what super easy to do, whereas computing this analytically is hard. So the question is, if you have a process-- if you have a program from which you can only get stochastic estimates of this, we might be interested in dL of P by dP, but how are we going to get that? Because we can't-- we're no longer-- we're not going directly from here to here. Somehow, we have to start here at X of P and get dL by dP. So this is one example.

Let me give you another example, which to me, is the most compelling example. Now, it's sort of already set, which is in the physical sciences, your model may be inherently stochastic. Your model is stochastic.

For example, you might have two molecules or two particles, and they're interacting with a rate r. They're binding or something that. But this is the average rate of interaction.

Whereas in reality, if you let these molecules roam around, they're going to interact-- the times at which they're going to interact is going to be a stochastic process in time. For example, in this case, if you have a rate r of interaction, then the times at which they interact, that's called a Poisson process. And if you say, how long am I going to wait until they first interact, that's actually distributed as an exponential distribution with rate-- with scale r, or maybe 1 over r. I'll have to think about that.

It should be 1 over r rates, because the higher the rate is, smaller-- the less time you'd expect to wait. But the main point here is that your model of the system-- you're interested in some physical process, and your model of it is inherently stochastic. So here, there's no getting around the fact that you have to deal with stochasticity.

You can't just switch to a different model that doesn't have stochasticity, because then you're tied to this physical process. So this is another case where you have to deal with stochasticity. And now if you want to-- let's say you have some model of what's going on, but you don't know your parameters. For example, you don't know r.

Then you want to fit your parameters to data. And this is, again, some sort of optimization or inference problem. So you have some real world data and you want to optimize the parameters of your model to best fit the data.

This is some optimization problem, and again, we're dealing with the same thing, which is have a system where it's very easy to simulate it in a stochastic way, but maybe it's really hard to exactly compute the statistical quantities for the system. So again, you want to solve the same problem, which is that we're interested in the derivative of the expectation of your program with respect to P. OK, so our derivative is not-- this doesn't mean-- this isn't our answer to the derivative, but it's something we'd like our derivative to help us compute, like if we're happy with many different notions of the derivative. But at least let me compute this using the notion of using the derivative.

And it's also a bit ambitious to hope to get this exactly analytically, because we couldn't even get the original expectation analytically. So a more reasonable goal might be-- so this is getting into automatic differentiation. It's like you're given a program X of P, and your goal is to produce a new program, which I'll call X tilde of P, which averages to the derivative of-- sorry. My screen turned off. Which averages to the derivative of the average of your original program, which is a bit of a mouthful.

So in machine learning, this is often called a gradient estimator that you're trying to find. And this isn't necessarily-- even X tilde of P, this may not be our notion of derivative that we're looking for. But it's certainly something we'd like to be able to construct from whatever notion of derivative we come up with. So this is the motivation for why we might want to characterize the sensitivity of a random program, why we might want a derivative.

PROFESSOR: Is everyone clear on this at this point? So this-- you're going to try and find a random program whose average is the derivative of the average of the original program. All right? So it's kind of a little mouthful. OK, people are nodding their heads. So--

GAURAV ARYA: OK. So that is the why we want to do this. And now the question is, how? What notion of derivative should we come up with to let us do this? So yeah.

So once again, the functions we're interested in look this. P maps to X of P. So now let's think back to the beginning where we had these two questions.

One was, just think about the differential, or just how the output is changing with respect to the input, without worrying about turning it into a derivative that cuts out higher order things. Just, how does the program-- how do perturbations of the program look like? So let's try and answer that first. Let's answer question 1 first.

So the first thing I'm going to write is-- just try to naively characterize the change in my program. So here, I'm using slightly different notation from before, and maybe Professor Johnson or Professor Edelman might object to this. But this is what I'm calling the differential for a stochastic case. This is a stochastic differential, and I'm writing it as--

So basically, epsilon is your dP. So this right-hand side should make sense. It's basically X of P plus dP minus X of P.

One thing I've done here is I'm not super comfortable with dealing with differentials as infinitesimal things, so I've written this left-hand side as a function. dX of epsilon equals X of P plus epsilon minus X of P. So this is just exactly what it says. There is no dropping of higher order terms here.

And epsilon, I'm thinking as-- so remember P was real. So epsilon, I'm thinking, is some small real number. So this is basically just, for a finite perturbation in the input, how does the output perturb?

PROFESSOR: Yeah, so this is--

GAURAV ARYA: Although I might not have made it totally clear what this means yet. That'll come next.

PROFESSOR: Yeah. So in lecture--

GAURAV ARYA: But I'm interested if--

PROFESSOR: In lecture 1, this was what we called-- in lecture 1, this is what we called delta X. So delta X--

GAURAV ARYA: Yeah, OK.

PROFESSOR: --was the nondifferential change. But that's fine. You can stick with your favorite notation. Don't change notations midstream.

GAURAV ARYA: Sure, yeah. So yeah. But this is-- so this idea-- but then, I guess, now you have to ask, what sort of object is this dX of epsilon? So maybe this is also something I can ask the class. X of P was a random variable and so forth, so what is the dX of epsilon?

PROFESSOR: So what is it? Yeah?

AUDIENCE: A random variable.

PROFESSOR: It's a random variable is the answer.

GAURAV ARYA: Yeah, yeah. Generally, if you subtract two random variables, what you're going to get is another random variable. However, I'd argue that I haven't really fully specified to you what dX epsilon is.

You're right it should be a random variable, but I think it could be a whole lot of random variables dependent on something I haven't really told you. Does anyone see why-- is it totally unambiguous, or-- let's think of one-- let's think of a concrete example. Let's think of the case where X of P is distributed as the exponential distribution with scale P. For this case, is it unambiguous what dX of epsilon is, or is it unclear?

PROFESSOR: So what's-- is something missing from this definition? Yeah? Someone's raising-- yeah?

AUDIENCE: I guess by "change," do you mean a CDF change? Or, like-- I'm not really sure what you mean by "subtracting random variables."

PROFESSOR: Yeah. So the question is-- the comment was he wasn't sure what you mean by "subtracting random variables." Do you mean the CDF change? Or what is this minus?

GAURAV ARYA: That's a good question. So what I mean is you-- so a random variable is basically telling you how to sample from something. A random variable says how you-- it gives you the procedure for sampling from a distribution. So I'm saying, take a sample-- I'm going to construct a new random variable which says, how do you sample from dX? You sample from X of P plus epsilon, you sample from X of P, and you subtract so that is-- yeah.

PROFESSOR: It's a sampling procedure. OK.

GAURAV ARYA: So it's very different from doing operations with distributions. If you were to look at this-- if you're trying to figure out the distribution of dX in terms of the distribution of these two, when you add two random variables, the distribution convolves. You convolve the distribution.

But if you think sample-wise, then it's just a simple subtraction. Yeah. But why might this still not be fully specified?

PROFESSOR: Any other ideas what might be missing? Yeah?

AUDIENCE: Is P Fixed, or--

PROFESSOR: Is P fixed? I guess, is that--

GAURAV ARYA: P is fixed. Actually, I should have said that. So that's a good point. So in some sense, there's a dependence on P here that I'm sort of suppressing.

Imagining you consider a fixed P, and now you're thinking about a neighborhood of P. But yeah. So that is true. Yeah.

PROFESSOR: Any other ideas? No? We're drawing a blank, Gaurav.

GAURAV ARYA: OK. So we know what's so-called the marginal distributions of each of these random variables. We know how to sample from each of these. But there's a question of what their joint distribution is, which means-- for example, if you have--

Suppose you were considering X of P minus X of P for a second. Well, maybe that's a bit too confusing. Let's say we're considering-- so let me go to a new page. Let's go away from that context for a second and let's consider subtracting two random variables, A and B, where let's say A and B follow the same distribution.

So A and B follow the same distribution. Let's call it some distribution d. Is this going to equal 0? Like-- If the answers are yes--

PROFESSOR: So if A is a-- if A and B are both coin flips with probability 1/2 half they're 1, and probability 1 they're 0, is the difference 0? People are shaking their heads no.

GAURAV ARYA: Yeah. So it actually depends. Because if A and B are independent, like you would imagine for two coin flips, then it would not always be 0. If on the other hand, A and B could be the same thing, they could literally equal A, in which case, yes, this would be 0.

So this is the idea of a joint distribution. Even if you know how to sample from a random variable marginally, you can figure out, in the whole probability space, do they-- how do they behave together as they-- so in some senses, what we haven't specified is a distribution of A comma B. Does that make sense?

PROFESSOR: Yeah. So are they the same coin flip, or are they independent coin flips, or something in between? Yeah. People are shaking their heads yes, so I think--

GAURAV ARYA: OK. So let's now go back to our particular case of subtracting two random variables where we had X of P plus epsilon minus X of P. And my claim is it's not enough to say that this thing is distributed as an exponential distribution with rate P plus epsilon, and this thing is distributed as an exponential distribution with rate P. This doesn't tell you enough.

You have to specify how they behave jointly. And perhaps the most simple thing to do is to say, why don't we make these two random variables independent? So basically, it means these sampling processes are not tied to each other at all. We're going to take a sample from a sample from X of P, a sample of X of P plus epsilon, and subtract them.

So let's actually see how that works in code. So yeah. So my sampleX is already set up as an exponential random variable, and now I'm going to sample_dx, which takes as input epsilon. This is a little confusing. Let me put that. And I'm going to fix p at 100. So then this looks like sampleX of 100 plus epsilon minus sampleX at 100. And now let's try sampling from dX just to get a feel. So you can sample dX. Maybe let's assume a small perturbation, like 0.1.

And we can see it looks like a random variable, as was said. And it takes fairly large values. Here's a negative 145. Maybe let's make epsilon really, really tiny. It still takes fairly large values. It still looks pretty large.

And at this point, maybe alarm bells should be ringing because in a derivative, you sort of have this idea that if you perturb the input by a smaller and smaller amount, the output, in some sense of size, some sense of norm, should get smaller and smaller and smaller. Whereas here, this isn't happening at all.

So if you were to try and do something like divide by your step size, as you often try to do with derivatives, this looks concerning because, sure, this might average to the right thing. But its variance is going to be really large because the right thing might be something like, I don't know, 1. And we're getting these numbers of order 10 to the 7, which, if you trust the math, are going to average to 1. But if you actually want to do this computationally, you're in trouble.

So in making all of these, making these independent is not a good idea. Are people convinced by this? Are there any questions?

PROFESSOR: So I think people are shaking their heads OK.

GAURAV ARYA: OK. All right. So what could be a better approach to this? Well, here we have to-- so let's again stick to our example. And all these papers, I'm going to fill up all their nooks and crannies later. I feel bad about wasting so much. Yeah, so let's go to this example. And now we have this family of random variables.

And we need to figure out how they work with each other. How are they distributed jointly? And here I have to introduce some more concepts. So how is a random variable actually defined? We're trying to differentiate these objects. Maybe we ought to understand these objects a bit better. So this is getting into a bit of probability theory.

But basically, there's something called a sample space. Omega. This is a space of things. And it's equipped with a probability distribution p.

Some other way you can think about this is there's some random variable with output space omega. But here I've actually written the distribution. And then your random variable. Let's say any x of p is a map. And also, for simplicity, I said p was real. Let's also make x of p a real valued random variable, for simplicity. In which case, x of p is defined as a map from omega to R.

So this is a bit confusing because this is like a deterministic function. So how are we defining random variables through a deterministic function? The idea is that all of the randomness is here. This is a distribution which is independent of p. This distribution doesn't depend on p.

And then we have a fixed distribution, a fixed source of randomness, for all of our family, all of the random variables in our family. And what differs within our family is what the map is. This map is parametrized by p. So x, you can write x of p on the main--

PROFESSOR: It might help to be concrete here, Gaurav. So on a computer, there's a function in Julia called rand that will return a number between 0 and 1 uniformly distributed. So now suppose you're given that as input, and then you want to spit out an exponentially distributed thing as output.

So you only need that source of randomness, which who knows where that comes from? And then somehow, it passes through some map to produce some other distribution, other type of random number.

GAURAV ARYA: Right. Yeah, yeah. So actually, yes, I'll try and show that for an exponential distribution. Are there any other questions first?

PROFESSOR: OK, no other questions right now.

GAURAV ARYA: Yeah. Yeah, so I'll try and show that for an exponential distribution. So again, x of p distributed as an exponential. And then for simplicity, just for consistency, this isn't actually the simplest choice for doing this for an exponential random variable, but I'm always going to make omega between 0 and 1. And p is the uniform distribution over 0 and 1.

PROFESSOR: You have a coin flip.

GAURAV ARYA: Yeah. Exactly. And sort of going back to this map thing, maybe I should have said also explicitly how do you sample from x of p? And the answer is sort of like a two-step process. First you pick a random omega, according to p, and then you plug it into this map. That's how you sample from x of p. So here this is like your call to rand. And this is going to give you-- if you think of sampling from p, that's going to give you an omega, which is between 0 and 1.

And then it turns out that the function for an exponential distribution, think it's log, maybe negative log. It should be negative log. The negative log of a uniform.

And I could also just write omega here. I'm writing 1 minus omega. But this is x of p of omega. So maybe it'll become more clear if I try and show you in code.

PROFESSOR: Questions?

AUDIENCE: Shouldn't that depend on p in a way?

PROFESSOR: The question was, shouldn't that depend on p? Oh, yeah. Oh, your p scaling, yes. Yeah, you didn't put your scale in.

GAURAV ARYA: Yeah, sorry. That should have been multiplied by p. Thanks. Thanks for the question, yeah.

PROFESSOR: Yeah, yeah.

GAURAV ARYA: Yeah, so we add--

PROFESSOR: So there's a p outside the log. So that's [INAUDIBLE].

GAURAV ARYA: Yeah, so if you go back to here, we had a way of sampling from an exponential distribution. Now let's write our own sampler, where we use this trick, where our fundamental primitive is going to be rand. So maybe I'll write this in multiple lines. So first, we sample our omega, which is a uniform random number.

PROFESSOR: Yeah, so rand is uniform between 0 and 1, basically.

GAURAV ARYA: Yes, yes. And then we return log of 1 minus omega.

PROFESSOR: Times p.

GAURAV ARYA: Times p, yes. So yeah, so now if I try this with 100, it does seem to look something centered around 100. We can check its mean. So yeah, so, in fact, under the hood, this is probably what your computer is doing. There's a really optimized way of generating uniforms. Actually I'm not 100% sure that's how it's doing it, but this is one possible way. And why is this relevant to our differentiation problem?

It's relevant because if we go back to this phrasing, the key part is that this call to rand is independent of p.

So what we can do is we can not make all of these x of p's independent, but rather make them all rely on exactly the same call to rand, which we just plug into their respective maps. Does that make sense? Since I'm a bit short on time, maybe I'll switch to some slides.

PROFESSOR: There's a famous quote in computer science that random numbers are far too important to be left to chance. So it's quite difficult to get a source of randomness on a computer. So there's a lot going on in the rand function. So once you have that, you don't want to touch it. And then you kind of reuse that for everything else.

GAURAV ARYA: Yeah, so this is how it looks pictorially. So we're sampling this random number from omega, and you're plugging it in to x of p. And this is your map. And then you're sort of looking it up on the y-axis. OK, so that's how x of p looks like. We sample a uniform from 0 to 1, we plug it in. And at a point is that x of p plus epsilon, for instance, is jointly distributed with x and p.

So if we were to pick this omega 1 on the real line, then our sample from x of p would be this. I hope you can see my mouse. And our sample from x of p plus epsilon would be this. And then their difference would be our sample from the x of epsilon, which, as you can see, is sort of getting smaller as epsilon gets smaller.

So this whole procedure, which I'll describe in a bit more detail, they're called the reparameterization trick, and it's a very old trick and used nearly everywhere. For example, in those variational autoencoders. So before the reparameterization trick, you can also imagine how our idea of sampling from them independently would look like. And it looks like this. Rather than using the same omega for both of them, you sample two independent omegas, and you take their difference.

And you can see that this can be pretty huge, no matter how small I make epsilon. So it's sort of like you're trying to figure out how this curve is shifting, how fast the area under this curve is changing. And the way you decided to do it is like a random sample from each of these curves. But what's going to happen is the noise of the intrinsic randomness is going to totally wash out the signal of how much my curve is going up.

And the solution to that is use the same value in the y-axis, same omega, for both curves and directly sample their difference. That's how you can estimate how fast the curve is moving. So that's what we're going to do. So this is your dX of epsilon, which sort of subtracts these at every fixed omega.

And now the nice property is that at every fixed omega, dX of epsilon is kind of like order epsilon. It's of magnitude epsilon. It's not O of 1. I think you've used this big O notation in the class.

PROFESSOR: Yeah, we used little O notation, but yeah.

GAURAV ARYA: OK, OK.

PROFESSOR: This is similar, yeah.

GAURAV ARYA: Yeah. Yeah.

PROFESSOR: So it's proportional to epsilon or smaller.

GAURAV ARYA: Yeah, so this is your differential. So we've solved problem one, which was how do we describe a change in our program? We've decided that it's going to be itself a random variable. And it's basically the subtraction of these two random variables with a particular joint distribution. Namely they're all sharing the same source of randomness. Yeah.

And the final piece of the puzzle-- that's a bit scary-- is now that we have differentials, we can just take the derivative kind of like point-wise at every point on every omega. So we have this differential. And this differential, if you fix omega 1, it's moving up in a nice well-behaved way. This change is small.

So we can actually compute its derivative. And we can compute its derivative at every omega. And then we'll have a new random variable. So remember a random variable on a probability space is just a function from omega to some output. So now our function from omega to our output will be this derivative at every particular omega.

So that's represented by this thing here, the limit as epsilon goes to 0 of dX of epsilon over epsilon. And I'm calling this delta. And it turns out you can maybe-- it maybe it's pretty clear intuitively staring at this, that if you're sampling the derivatives at every fixed omega, what you're going to get, on average, is a derivative of the average of the whole program.

I said that our derivative might not be exactly the same thing as the gradient estimator, but it sure looks like, in this case, it is. Like our derivative, which tells you every point in the sample space how much is the random function changing, is just, if you average it out, it is giving us what we decided we wanted at the beginning. So that was quite a lot. Are there any questions?

PROFESSOR: Any questions at this point? So everyone's silent so far. Oh, one question.

AUDIENCE: What does it mean for the derivative to be 0?

PROFESSOR: So what does it mean for the derivative to be 0 for a program like this?

GAURAV ARYA: For the derivative to be 0. So there's two senses. So yeah, the derivative to be 0 everywhere. So delta is a random variable. So maybe I should also just state for you what delta is for an exponential random variable. So let me show you that.

So where did I have my-- I lost it. It's over here. OK. So this was our function. There was a little p added in there. Negative p times log of 1 minus omega. So we had the dX-- I'll need a new page. So dX of epsilon. This is a random variable, which, given a probability space, is just a map, taking an omega.

And it's d by dp of negative p log of 1 minus omega, which is equal to negative log of 1 minus omega. Sorry. Sorry. This is delta of omega, I would say, instead of taking the derivative. So delta of omega is negative log of 1 minus omega. So this is your random variable. So in this case, it certainly looks like it's not always 0.

Although interestingly, it's always constant because this p was kind of just the scale. Now your question is, what if we got a case where delta everywhere in omega is 0? Well, that would just mean that your random variable isn't changing at all. So except in very degenerate cases, I wouldn't expect that to happen, where the whole distribution of your random variable isn't changing.

Perhaps slightly more common would be a case where the derivative of the loss, the derivative of the expectation, is 0. Maybe it's slightly more common. And that would happen when the expectation of delta is 0. Yeah. Does that make sense?

PROFESSOR: Another question.

AUDIENCE: Yeah. I was just curious if this had any connection to what it means to be a stationary distribution for the exponential.

PROFESSOR: So the question was, does this have any connection with what it means to be a stationary distribution?

GAURAV ARYA: To be a stationary distribution. Not that I can think of at the top of my head. Yeah. I'm not totally sure. I mean, I guess there's some notion of derivative being 0 there.

I guess if you let your p be sort of like timed, like sort of like a stochastic process in time, and you're differentiating with respect to t, then I would expect even then, even if you're-- I mean, yes, so it could be the case that you get-- you would definitely get this if you're at a stationary-- if your t is approaching infinity, so instead you're short of reaching a stationary distribution, and you're doing x of t and x of t plus delta t, then any statistical quantities are going to go to 0.

It's possible that this is still non-zero because the parameterization depends. I really have to think about it more. But yeah, maybe there's some connection if you think of t as a time or number of steps in some continuous sense. Yeah.

PROFESSOR: Another question?

AUDIENCE: Shouldn't that be minus 1 over p?

PROFESSOR: Shouldn't that be minus 1 over p? Where?

AUDIENCE: No, like--

GAURAV ARYA: Oh. I don't think so. I mean, actually I think there are two different conventions. That's why it's very important that we clarified at the beginning that t was a scale. Often the exponential distribution is parametrized by the rate, which is the inverse of the scale. But here this makes sense to me because it's getting larger with larger p, which would make sense [INAUDIBLE].

PROFESSOR: Yes, p is not a probability.

AUDIENCE: [INAUDIBLE] 1 minus p to the power minus px or minus x by p?

PROFESSOR: Yes. So the point is p is the scale of-- so the average is p of x. All right.

GAURAV ARYA: Yeah, so maybe I can write down our gradient estimator program. So this one looks like delta. And actually, it's technically a random variable, but in this case-- oh, sorry, it is a random variable. Sorry, I misspoke when I said it was constant earlier because it is a function of omega. So this is our random variable as a function of omega.

And then if you want to sample delta, we can do negative log of 1 minus rand. So theta and rand. And now if we sample this a bunch of times, I'm getting something close to 1. And it actually turns out-- so if we go to our-- so let's just remind ourselves what x was. x was this.

If I do this, this is actually p. So the derivative of p-- the derivative of expectation, which is p with respect to p, is just going to be 1. This makes sense. But then the crucial part is if we're interested in automatic differentiation, this notion of derivative-- we could have just said our notion of derivative is 1. That's the expectation of x by the p. It's 1.

But the issue with that notion of derivative is it doesn't compose. So consider this new function. And remember it's super easy to just make the distribution really complicated by just doing a little bit of extra stuff. So here I'm sampling from an exponential distribution. I'm squaring. And it has some mean. And I might be interested in its derivative. And what you can do, I guess you've learned about dual numbers. So this is going to be a bit fast. Or maybe I'll just handwrite it.

So if you use the chain rule, it should be 2 times the sample from x times a sample from delta. And you're just going to have to trust me that that's the true derivative. I'm not 100% sure because I did a bit of-- I'm not 100% sure I got it right.

But basically, the chain rule just works with this notion of derivative, which isn't something I've totally justified to you, but that's sort of the upshot. So that's why this is really popular. This is a really popular trick. You can also use it for reverse mode differential.

PROFESSOR: Gaurav, shouldn't that have been the same omega in the x, 100, and the sample delta? Right now, you're using independent random numbers in x, 100, and sample delta in that 2 times x times sample times delta.

GAURAV ARYA: Yeah, you're right, so this is probably not the right answer. So I would have to do--

PROFESSOR: So they should have really been the same random number.

GAURAV ARYA: --delta of rand. And then write my x as like negative 100 times log of--

PROFESSOR: No, it has to be the same--

GAURAV ARYA: --1 minus rand.

PROFESSOR: No, no. No, you're still using independent random numbers.

GAURAV ARYA: Oh. OK. Well, OK, maybe I don't have time to get it right here. I guess it would be an omega and a rand. But the point is that this notion of derivative obeys the chain rule, which is really nice.

PROFESSOR: Yeah, because basically you're differentiating that it's just the ordinary derivative of that completely deterministic function that takes in rand, omega, and outputs something else. Once you have the deterministic function, it's the ordinary derivative, ordinary chain rule, ordinary everything.

GAURAV ARYA: Yeah, yeah. And I think if you search up reparameterization trick, because it's gotten very popular in machine learning recently, there's a lot of resources you can find online. Some of them may explain it in a more barebones, simpler way than what I have done because part of the reason I've done what I've done is also to think about how we might generalize to the discrete case, which I'll try and just maybe just do a five-minute overview of, if that sounds fine?

PROFESSOR: Sure. Yeah, that's fine.

GAURAV ARYA: OK, so I'll just try and briefly explain. So basically, my project has been about taking that idea and now handling discrete random variables. The other example I gave at the beginning was a Bernoulli random variable. And now we want to apply the same approach. Come up with a notion of derivative that sort of explains the sensitivity of our program with respect to p. And why is this more difficult for a discrete random variable?

So does this step function here make sense? Basically I'm using the same probability space. I'm choosing a uniform omega from 0 to 1. And now this point here is 1 minus p. So if I land on the right, if I land in this region of probability p, I'm going to get a 1. Otherwise, I'm going to get a 0. Does that make sense?

PROFESSOR: Yep, people are nodding yes.

GAURAV ARYA: All right. So now if we perturb p, something qualitatively very different happens, which is that where the step occurs changes. So if you think about it, nearly everywhere in our sample space, nothing is changing. So as I make epsilon smaller and smaller and smaller, that object delta now that I defined before is just going to be 0 everywhere because for small enough epsilon, nothing interesting is happening.

But does this mean that the derivative of your expectation is 0? No. In fact, for Bernoulli, the derivative is 1. So what are we missing? We're missing the fact that the difference, if we look at the differential between these two, it's nearly everywhere 0. But in the place where it's non-zero, it's huge. It's a flip of the whole-- it's like a flip of the coin. This 1 isn't getting smaller and smaller in epsilon.

So for people who like analysis, there was this interchange of limit and expectation that occurred here when we were trying to justify that this reparameterization trick works. And that is no longer valid, in this case, because if you think about dividing this by epsilon, this magnitude of this thing is really, really large. It's 1 over epsilon. So that's the essential challenge of discrete randomness.

And if you think about it, so remember the derivative is the best linear approximation of the sensitivity of your program. It's the best linear approximation of how your program gets perturbed. And here what's linear is how fast this step is moving to the left.

So it kind of feels like we want to differentiate not the value, the change in the value, but rather we want to differentiate the probability that they're different.

So that's sort of the idea of how to extend the usual reparameterization trick, which just has delta, into an object which can also capture these flips with tiny probability, which you see in the discrete case. So I know that's very fast. And it's just really just the essential idea. Do people have questions?

PROFESSOR: And so Gaurav actually just published a paper on exactly this. Basically you write a program that has rand calls and maybe has an, if rand is greater than 0.5, return cosine. Otherwise, return sine or something like that.

And it can differentiate that in this sense, give you something whose expectation value is the derivative of the expectation value. But you have to track these discontinuous things that don't have an 18.01 derivative. You have to track them separately, in some sense.

GAURAV ARYA: Right. Yeah, so here's a quick demo. So here I have a Bernoulli, which its mean is around 0.5. So we'd expect its average value to be-- so we'd expect its derivative to be 1 because the expectation is p. So again, I'm always doing this boring case where we just have a single distribution. Maybe I'll try something more complicated right after this. But basically, the idea of our approach, using dual numbers-- this looks like a dual member, right?

There's this epsilon thing, which is like this infinitesimal change. So it's like you got a 1, and your derivative is 0. This is sort of what the reparameterization trick would always say, that the derivative contribution is 0. But sometimes, what these triples tell you is that you could flip with probability that's like something times epsilon. So it's a probability that's differentiable.

And then what this would be telling you, if you collapse this into a derivative estimator, so now the derivative object, which is all of these numbers, is no longer the same thing as your gradient estimator. But it can be collapsed into your gradient estimator by doing this value plus this value multiplied by that value.

So if we sort of maybe-- if we let this be our triple, you can sort of collapse it into a gradient estimator. And it's going to be 2. But it's like 2 50% of the time and 0 50% of the time. So on average, that's what? Which is cool, but it's not very impressive for a single Bernoulli. So maybe I'll try a random walk. So it has to depend on some parameter p.

So I'm going to do a random walk where I start maybe at 0. I equals 0. Maybe n equals 0. And I take 100 steps. Maybe I'll zoom on it more. And each step, I'm going to either stay put or go to the right. So I'm going to do n plus equal to rand of Bernoulli. And then I'm going to make my transition function depend on p.

For example, in this case, maybe I'll do something like p over 100. Yeah, p times i over 100. So this is something that depends on where-- or maybe that should be [INAUDIBLE]. This is something that depends on where you currently are in your walk and also depends on p. And if this Bernoulli turns out to be true, you're going to walk to the right.

Otherwise, you'll stay put. And then we're going to return n. So now we can just check that this function runs. It might not. Or it might do something strange. Oh. Yeah. Oh, this probability becomes tidy. Let's make p a bit larger. So this is like p of 50.

PROFESSOR: It's always 0 because n starts at a 0.

GAURAV ARYA: Oh. OK, thank you. Maybe I'll do 100 minus-- so let's do 1 minus all of that. Does that sound reasonable?

PROFESSOR: Yeah.

GAURAV ARYA: No.

AUDIENCE: [INAUDIBLE]

GAURAV ARYA: OK. So let's have a function which goes from 1 to 0, dependent on where you are. So it's going to start at 1, go to 0. And let's multiply it by p. And then let's make p something like 0.5. OK, something interesting is occurring. So we just sample from this. Get something random.

PROFESSOR: Yeah, so other people were suggesting probability depending on i. But he's doing even more complicated. The probability depends on where you currently are. Right.

GAURAV ARYA: OK. Great. All right. So I'll do that too. So how should I do this? Maybe I'll do 1 minus n plus i over 200. This should be a valid probability. 0.5. OK. So it has a mean.

PROFESSOR: OK, yeah.

GAURAV ARYA: OK. If you just try and differentiate it the naive way by doing something like this and then maybe dividing by this-- have I got my brackets right? There's one missing, right? That should be there.

OK. Then you're going to get something that's pretty large, which you'll have to trust averages did the right thing. But now if you try and use these stochastic triples, so maybe let's just do one stochastic triple.

PROFESSOR: So again, the stochastic triple is the analog of what Professor Edelman did a couple lectures ago with automatic differentiation, where you had the value of the function and the derivative. We call that a dual number. Now you can run three numbers, the function, the derivative, the change up, down of the distribution, and also the probability of a big jump. So now you need a third bit of information. You carry through.

GAURAV ARYA: Yeah. So yeah, so for these stochastic triples, I'm just showing you the whole triple object first. There's never a discontinuous part. The deltas are telling you how your output is changing continuously. But your output is an integer, so it can't change continuously, but it can change to an adjacent value with some probability.

And the nice thing is even though the range of your function is pretty large-- it can be 36. It can be 29-- we sort of couple the original 29 to the alternate, which is like 30, which means that your estimator is going to be lower variance than average. It's just sort of the stuff I was saying with this joint distribution and what the reparameterization trick does so well is something we're trying to capture with these stochastic triples. It's still a work in progress, and there's a lot of issues.

But now we can also-- so now we can take this stochastic triple, we can propagate it through your program, we can collapse it into a gradient estimator. And then we can do this 1,000 times. And you're going to have to trust me that this is the derivative of your program. You could verify it by computing the transition matrix or using an alternative gradient estimation method, but that's the upshot.

That with this new notion of derivative, we can now answer this problem we wanted to solve from the start, which is for any complicated stochastic simulation, can we get a new program which averages the derivative of the average of the original? Yeah. That's all I had. Are there any questions?

PROFESSOR: Any questions? OK, so let's take a five-minute break. So we'll start again at 12:14.