GILBERT STRANG: OK, here we go. All set, and two topics for today-- one is to go back to Professor Sra's lecture. That was last Friday.

And he promised a theorem and proof. And this morning, he sent it to me. So it's proving the convergence of stochastic gradient descent.

And really, what's important, maybe, and useful is not so much the details of the proof, which I'm just learning, but the assumptions-- what's the logic here, what do you have to assume about the gradient and about the algorithm to get the answer? But now I actually look back at the video of his lecture. And it was excellent.

And as I looked at it, there were a couple of things later in the lecture that I thought would make good projects. So I don't know if anybody is still open to what to do on a project. But here are my two ideas.

And if you've already finished your project, well, you get an A-plus by considering one of these. So you remember-- and this will remind you of the lecture, which is a good thing. So do you remember that question 1 was whether, in the stochastic part, after you've sampled one or some mini batch-- but let's just say one of the lost functions, coming from one sample-- you remember, the whole point is that if we do all zillion samples at every iteration, we're really, really slow.

So the stochastic idea is to randomly pick one or a mini batch of the samples and just reduce their loss, just deal with the loss-- say, the square loss. Or later we'll see cross-entropy loss. But whatever the cost is, just do a few or one. And then the question was, after you've done that one, do you put it back in the pot every time you sample over the whole collection?

But that's expensive. Or do you just make a list of random order of all the samples and go through them? Which is then without replacement, which is a sort of semi-illegal.

That is, the logic in the randomization asks you to replace every time. But nobody does it. It

costs a lot-- probably not worth it.

So the project would be, suppose you take 1,000-- or, say, just 100. 100 random numbers-- say you use MATLAB, just the command "rand." So you get numbers whose average is a half from rand. They're between 0 and 1.

OK. So we know what the average is. So let's compute it two ways. One is by not replacing. And that's the interesting one.

So take 100 samples. Well, I guess we know that, after we've got through the full 100, we're going to get exactly the right answer. But anyway, my question would be, how much difference do you see in the eventual approach-- so the law of large numbers, I guess, would tell us we get a average of a half for these numbers with uniform distribution between 0 and 1.

Should I be writing anything here? Maybe I should. OK. So this is project 1.

You pick numbers ak, which is from rand-- so uniformly on 0,1. And then my question is, what about convergence to the final-- the average is a half.

So this may be too simple an example. But could we see what happens for the convergence of the average as you either do replacements or don't do replacements? And in fact, I would like to see a figure that looks like those in his lecture. Do you remember?

He started it somewhere-- start-- and then here's the finish. But you remember, the stochastic gradient descent was kind of pretty effective at the beginning. Well, the beginning, those might be 100 iterations each-- one epoch, one run through the full number.

But then when it got to here, got closer, it started oscillating. You remember, he identified the region of confusion around the thing. Well, my suggestion is just, I think those videos should be accessible to you on-- are they on Stellar?

Yeah. So I'd love to see that behavior and some good examples of that behavior and some pictures to you. So that would be one idea with and with-- oh, yeah, that's also idea 2. Idea 2 is the good start and then the bad finish for a stochastic gradient descent.

And of course, even without this, the magic words in computations is "early stopping." We don't over-fit. So we wanted to stop early, anyway.

And early stopping just is a good idea if that's what the approach to the x star that you're looking for. This would be the place where the-- that's x star where grad f at x star is 0. That's the minimum point.

That's ARG MIN-- exactly what we're looking for. And we don't find it very well. But we get close to it fast.

OK. Two ideas on projects-- so maybe I'll go to the main topic of today-- the topic I promised-- the idea of back propagation. This is all to compute grad f-- the gradient. All the derivatives-- this is the df dx1 to df dxm, maybe, I'll say, where I have m features for the sample.

OK. So that's back propagation. And that's the thing whose discovery, or rediscovery, put neural nets on the map. That's the key calculation, of course, to find the gradient. In the steepest descent algorithm, every step needs a gradient.

And if you can't compute it quickly, you're in bad shape. But you can compute it quickly by this automatic differentiation in reverse mode, which is otherwise known-- I don't think the people-- maybe Hinton was the leader in developing deep neural net-- deep learning.

So I give him big credit for that-- that back propagation would work and would give him fast gradients. But it actually had been studied before under the name AD-- Automatic Differentiation. So may I just tell you that idea? Some of you may know it, may know about it, may know more than I, and might know a good website to see this description.

There will be, of course, a section of the notes, you already have it. This is section 7.2. So this is the chapter on deep learning.

And the first section was about the structure of F of x. And you remember the key point about the structure of F of x is that I start with x and apply some function, F1 of x. And to that, I apply some function, F2 of x. And to that, I apply some function of F3 of F2 of F1 of x.

And that's the thing whose derivative I need. So I'll just take ordinary derivative-- well, partial derivatives, really. Yeah, I better say partial derivatives.

So suppose x is a pair, xy. Example-- so here, let me show you my example. So suppose F of x is-- let me take a simple example-- x cubed times x plus 2y. OK. So I want to think of that function the way anybody would, as the product of two functions.

So there is a product rule to get into the derivative. And then we need the derivatives of each piece. So there's a power rule and a linear combination rule. So it's got a few of the rules that we use.

And the point is to think about the computation of F of x and the computation of dF dx and the computation of dF dy. Those are the derivatives that we need. This is the function we need and how to do those computations quickly.

OK. And this is section 7.2, which benefited a lot from a blog. I'm not a blog reader or a blog writer. But somehow I found this blog.

It's Christopher Olah, is his name. And he really writes clear things. He works for one of the big companies and does the deeper research. But he's also a really good expositor.

And the website that he now uses is called Distill dot something. But I think maybe this blog was earlier than before the start of Distill. But it might be loaded onto Distill.

Anyway, that's where I got this simple description of back propagation. And let's just do calculus, first of all. If I just have a function of maybe even one variable, what's the derivative?

What is dF dx here, just to remember what calculation we have to do? So dF dx, this is with n equal one-- one variable. So I use ordinary derivative and not partial derivative. But that's what really has to be done.

But just, what's the derivative of that-- of a chain of functions? Well, of course, the chain rule. So what does the chain rule say?

I differentiate dF. I don't know. What do I put that it's differentiated with respect to? dF3, dF2-- is that what I should put? OK. And where do I evaluate that derivative? So yeah, I don't evaluate it at x.

I'm differentiated to F2. So do I evaluate it at F2 of F1 of x? This is where the chain rule gets sort of a little chain-ey. OK. Then we know that dF2 dF1. And again, that's now evaluated at F1 of x.

And then the final factor is dF1 dx evaluated at x. That's somehow what we have to do. And that's just for an ordinary one-variable function.

And I have here a two-variable function. And deep learning has a million-variable function. So I

think we won't go to a million. But two, we could manage.

So let's compute the function, first of all. Compute F. So I'm given x equals, say, 2, and y equals, say, 3. And I'm going to create a computational graph.

So I'm actually going to draw the computational graph to compute for F. And then it'll be a variation of that graph to find the derivatives. So let's just start with the graph, first of all, for the function, because we're going to need that.

So again, it's x cubed plus-- so can I write that function again? x cubed times x plus 2y. So I think the first step will be to find x plus x cubed-- that factor, which will be 8. And we have to find the other factor, x plus 2y.

So then that uses y and x. So it's a directed graph in going forward with this computation. So x plus 2y equals whatever it is-- 2 and 6-- oh, 8 again. Not brilliant. What shall I change here? Make it 3y?

3y, just to get a different number here. So now x is 2. y is 3. I get 11. That's a good number. 11.

OK. So far, so good? And now the next step on this graph will be, I have a product of those. So that will go to the product.

F equals 8 times 11-- 88. OK. So we've got the answer, 88, which, normally, I wouldn't take that much of a book to compute F. I would have said, 2 cubed times 2 plus 3 times 3. And I'd have simplified that to 8 times 11. And I would have got 88.

So if we were just writing normally, that would do it. But this is the picture of the computational graph. OK. Good. Good. Good.

Now it's the derivatives-- two derivatives to find-- dF dx and dF dy. Suppose we go forward first. My point is going to be-- or the great point is that backward is better. Reverse mode is better.

But we don't know what that means until we've gone forward. So let me go forward. So now I'm going to go forward.

Let's do dF dx. Everybody is up for dF dx-- the partial derivative with respect to x? So here we have x equal 2 and y equal 3. OK.

And then I take the derivative of that step. The first step was x 2x cubed. So I need the derivative. The whole point of AD is that every computation of a derivative breaks down like this into very simple pieces.

And the derivatives of those simple pieces are also simple pieces. So the whole point is to replace appropriately those intermediate steps with derivatives, so as to compute the x derivative. So I have to use the fact that the derivative of x cubed, with respect to x-- oh, I better do partial derivative-- partial derivatives of x cube, with respect to x, is 3x squared.

I'll put maybe a formula and then a number. So that gives 3 times 4-- 12. And the derivative of x cubed, with respect to y, gives 0, clearly. So that's 0.

So I'm doing the x derivative. So the derivative of y, with respect to x, is-- you get to tell me. If I'm computing partial derivatives, it is 0. It is 0. y and x are independent.

And this is the reason, in my view, that the forward method is wasteful, because I'm going to have to do another whole graph for the y derivative. In other words, tracking the x derivatives, a whole lot of stuff never got off the ground. So we never should have looked at it.

So anyway, I have this x plus 3y, maybe. I don't know whether to erase that. I think I will, just because I don't know what to do with it there. Yeah. So now let me take the ones that I really need, is the derivative, with respect to x, of x plus 3y, which is 1.

And so that gives me the answer 1 for any x actually. OK. And now what? Oh, yeah, I don't need these. This is a waste of time. Isn't it? Is it only x derivatives I want?

Anyway, let's just keep going. You can see, this takes a little organization. And I'm not practiced with it.

So what am I going to do? I'm looking for the x derivative of-- I've got to use our product rule now. I found the x derivative of that factor was 12. The x derivative of this factor is 1.

And now the x derivative of the product-- so now I'm going to do, somehow, a product rule-- the x derivative of this product. I should have given these two terms a name. Let me call that first term x cubed, and the second term x plus 3y-- call it s. So I'll call the two terms c and s.

So that's dc ds. This is dc dx. This is dc dx. And this one is ds dx and dc dy. Do I need to know that? I'm sorry, this computational graph has thrown me.

But now I want to use the product rule. And I'm taking x derivatives. So I should have computed c and s.

Yes, I see I need those in the product rule. So I should have computed c as being 8 and s as being 5. Is that right? 2 plus 3-- so 11. Yeah, I needed the 8. Oh, is that-- what's up?

I've just been running along here without getting myself in the whole picture. Yeah, 8 and 11 is right. But now I'm looking for the derivatives. So I don't multiply those. That's not the product rule.

So the product rule is what? So this product rule, I have to do this combination of-- this is now the product rule-- for the derivative of c times s. So I want c ds dx plus s dc dx. I think I'm on track now.

And now I want to put it in numbers. So c is 8. ds dx-- have we computed ds dx? Yes, ds dx is 1.

And now s itself is computed as 11. And dc dx, we computed as 12. I don't dare look.

I don't think I'm going to get-- oh, no, I don't know the answer yet. Sorry, I'm not trying to get 88. You guys are not helping. [LAUGHS]

You see I'm in trouble. But what I imagine here is, that's 8 and that's 132. So I'm getting 140. Is there any possibility that that's the right answer for dF dx? This is dF dx I computed.

By watching me struggle here, you're seeing the idea. Every step, I take the derivative of each step. So it was a power step, x cubed. So I had a 3x squared. And a sum step, so I had a 1.

Then the next step was a multiplication. So I needed the product rule for that. I have these separate numbers. So I put them in.

And so it's the computational graph finished. We only needed two levels. And we got 8 and 132-- 140.

OK. But we didn't get dF dy yet. And for that, I'd need to redo this again. And I don't want to do that.

I would rather do the reverse mode and do them both at once. That's the point of the reverse mode. It's very efficient. It's very efficient, actually.

Computing the gradient after you've done the work for the function, computing first derivatives-- you could compute n first derivatives with about four or five times the cost, not n times. That's amazing to me. That is amazing that I can compute the gradient very efficiently by the back prop.

So I have to show you the backwards way. Yeah. I'm just going to follow all the paths backwards so that I get both dF dx and dF dy. You see, the idea is to take the derivative of each step-- each small step.

That's really what we do in calculus. If you think about the start of a calculus course, what derivatives do we actually know? Do we actually use F at x plus delta x minus F? What derivatives do we grind out?

We do the derivatives of x to the n. Every calculus book starts with x squared and finds the derivative of x to the n. Then you do sine x and cos x.

Then what others? Are there any more? e to the x-- good, e to the x. And it's the inverse function log. In freshman calculus, you always write ln, just to be out of date. OK.

And now that may be the list. Is it? And then the chain rule. Are there others that you actually do a computation of?

Actually, e to the x is defined by the property that its derivative is e to the x. And then you discover what log x has to be. And sine x-- how do you do sine of x plus delta x? Well, compare minus sine of x.

How do you find the hard way, once-and-for-all way? You draw a little unit circle and mess with some angles. And you discover that the derivative of the sine is the cosine. That's if you've defined the sine as a ratio of sides in a right triangle.

Of course, you could define it as an infinite series. And then you would be back to just using that. OK. So calculus does exactly what we're doing here-- finds all derivatives by the chain rule applied to a few ones that it has worked out in detail.

But tangent of x, we would use the quotient rule. Secant of x, we would use the quotient rule, 1 over cosine. And the products, we use the product rule.

So really, calculus tends to seem fairly simple when you look back to see what, actually, you did. And then integration-- what is integral calculus about? More or less guessing the answer.

You have to integrate f of x dx. So really, what you have to do is sort of think, OK, what had this derivative? What function had that derivative? And mess around and get it.

So really, it's a freshman course, I guess. OK. So where am I? Backward. Right. That's the thing still to do.

How does the backward system work? OK, I'll try my best. OK. So here is the big goal. Back-- so reverse mode AD. Right. And let me make myself a little note.

The little note is to give you another example where the order that you do the computations makes a big difference. And that's not obvious that it will. There are many things in math that you could do in either order.

And it seems like, logically, you've done the same things. So another, and simpler, example which shows how one way could be way faster than another way is when I'm multiplying three matrices. So I'm multiplying three matrices-- A times B times C.

And the question is, do I do BC first and then multiply by A? Or do I do AB first and then multiply that by C? And of course, I kept them in order-- in the order ABC.

But the order of computations can be different. You get the right answer both ways. But those can be completely, completely different.

One can be 1,000 times faster than the other. So that's just to show-- actually, it kind of connects to this. And there is also another-- so I'll do that, too.

So this is example 2, where this is meant to be example 1. And example 3 leads to something called the adjoint method in differential equations or in optimization-- in computing optimum and maximizing it. Yeah.

Really, the underlying reason it gives us speed-up is, it makes the right choice in a product of three things. Yeah. So it'll be enough to do example 1 and example 2. OK, let me go with example 1.

This is now back propagation. Finally, we got to it. OK. Well, I look at my notes is how I do it.

So the notes-- this is section 7.2-- does these computational graphs. And then here is reverse mode. So it starts over here with the-- so I'm going to use the chain rule. So dF dF is 1. And then I'm going backwards.

And of course, I have to use the right rule. So I have to use the product rule. And then soon I'll have to use these power rule and linear rules. So of course, no change there.

The change is that by going backwards-- oh, I don't know if I completed that sentence, that I could find 100 partial derivatives, if the function depended on 100 variables, in about five times the cost of one variable-- three to five times the cost of one. So you would expect 100 chain rules would cost 100 times.

But you see, we're reusing the pieces in the chain and just having a larger-- our chain is wider. But it's not longer. And it's not repeated.

Anyway, so here I'm going to use whatever it is-- dF dc and dF ds. And I'm remembering that-- yeah, OK. So dF dc is s, and dF ds is c. That was because F started out as c times s. It was the product.

OK. Then we've got to evaluate those. And I'll look again to see that I'm hopefully writing down some of the correct things. OK. So now what I've written down next is dF dc is 5. Or no, 5 on that example.

What is it here? dF dc is-- c is x cubed. So dF-- oh, sorry, dF dc-- yeah, I want s. I'm looking for s here. Yeah. I'm looking for s.

So I'm looking for s. And that's x plus 3y. Am I doing this well? I want, in the end, to get the derivatives with respect to x and y-- the whole gradient.

OK. I think we started right. The first derivatives is to write c and s. And then let me leave these boxes open, just to get the picture.

Then I'll need dc dx, dc dy, ds dx, and ds dy. I think that's right. Here, I had a product of c and s. So I had two derivatives.

Here I have c and s, each to differentiate. So have an x and a y derivative of x and a y derivative. And now it's just a matter of putting in those numbers and following the chain backwards.

Maybe I'm not going to put those numbers in, because if I didn't reach 140, you wouldn't believe in back propagation. And that would be an unhappy outcome.

So I'll leave you to put them in maybe. Or the notes have a separate example that you can see. But do you see the point-- that in the end, I'm going to find dF dx and dF dy from the chain-- from one chain and not from a separate chain for x and a separate chain for y.

To me, that's the point of reverse mode. It's a little bit of magic. But you see the steps-- the ingredient.

And some of you have seen this before and maybe know a better exposition. I found this blog by Christopher Olah clear. And these very simple things, you'll see, are clear in the notes. But maybe another blog brings out other points to make here.

It's not obvious, maybe, that I could have 100 variables and do the calculation in four or five times the cost-- four or five times being instead of 100. Yeah. But it's possible.

OK. So could I close today with this one? How could those be different? You're computing the same numbers, the same AIJ, BJKs, CKLs, and doing these sums. But it certainly is different.

So let's just do that. OK. I'll do it here. And then at the right time-- and I guess it'll be after Professor Rao on Friday and Monday, I'll come back to Professor Sra's short proof of the convergence of stochastic gradient descent.

The whole point is to show you what assumptions do you need. You need some assumptions on the gradient, some assumptions on the step size. And for a good proof, all the assumptions fit together, and, dong, out comes the conclusion. And the conclusion would be how fast it converges-- stochastic gradient descent.

So there's some expected things, because it's stochastic. We expect some assumptions about the mean and the variance to go into the proof. So you'll see that.

But maybe it's too much for today. So I'll come back to that. I might even put it on Stellar and just close with this.

So suppose A is m by n, B is n by p, and C is p by q. OK. How many steps does it take to find A times B times C-- the product of those three matrices?

Well, if I go this way, I have to do BC first. So BC costs-- how many operations to multiply that

times that? npq-- nice formula. npq. Why is that?

Well, I could say that the answer is n by q. And every number in there was an inner product of a row and column of length p. So I have nq inner products.

And each one costs p-- multiply, adds. So now I have BC, which will be-- so now I have m by n. Then I have m by n, which is the A times B by C, which is now n by q. That's BC.

This is A, BC. And this one costs-- what's the cost here? m by n, m by q-- by the same rule, it'll be mnq. Good. That's the first way-- A times BC.

Now, the second way is AB times C. Let me write in again, m by n, n by p, p by q. So now I'm doing this first-- so AB costs.

Tell me again now, what's the rule for the cost of a matrix multiplication? mnp. mnp. And then I multiply m by p-- that's AB-- times p by q. That's C.

So I have mpq. So I have that together with that, or that together with that. That sum-- those two or these two.

And they're different. And let's just recognize the most important example. Suppose C is a column vector-- C for column vector.

So q is 1. There's only one column. So if q is 1, this way did np-- let's just specialize to that.

So specialize to C equal a column vector, which means that q is 1. I only have one column. So then A times BC is versus AB times C.

So let's just figure that out when q is 1. So npq is just np. And mnq is just mn, where AB is m and p. Oh, that's a bad one. Disaster already.

Those are potentially two big matrices, multiplying a column vector. So here I've done a matrix multiplication. I never should have done that.

This is a matrix vector. It gives me a vector. And then this is a matrix vector.

So I get nice numbers here. But I get a terrible number for AB. And then I multiply that by C. So that's mpq. mpq.

So mp is factoring out. So if I write it as n times m plus p versus this one is m that's factoring

out times m-- no. Yeah. What's up here? Yeah. Sorry. What am I doing?

Yeah. Is it p that factors out from this one? OK. p times m plus n, I guess. Sorry. Anyway, the difference is--

AUDIENCE: I think it's mp times p plus q. [INAUDIBLE]

GILBERT STRANG: Shall I go over it again or write--? Let me do just this thinking again. If q is 1, if I go this way, was that my final total when q was 1? And that's this?

No. m factors out times n plus p. Let's just get that right. Oh, no, n factors out. Sorry, n factors out times m plus p. And this way was all these things.

AUDIENCE: Both the m and the p factor out.

GILBERT STRANG: Both the m and the p factor out. OK. Thanks. Times n plus q. n plus q was 1. OK.

The whole point is, we've got this horrible multiplication of three big numbers. And this only had two big numbers. So this is orders of magnitude faster than that.

And of course, you would have done the calculation. That way, you would have multiplied the column vector by a matrix to get another column vector. And you would have multiplied that by a matrix to get another column vector, where here, you crazily multiplied two big matrices together and then got a column vector.

So there is a bad move. OK, thanks. Oh, I'm past the time on this ABC. It's just to show that on a very familiar calculation, you have to do it in the right order. And back propagation is the right order for partial derivatives.

OK. Thank you. And so bring laptops Friday. And look forward to Professor Rao. Give him a good welcome.