Numerical approximation of gradients

Have a question? Discuss this lecture in the week forums.

Interactive Transcript

English

Help us translate!

0:00
When you implement back propagation you'll find that there's a test called creating checking that can really help you make sure that your implementation of back prop is correct. Because sometimes you write all these equations and you're just not 100% sure if you've got all the details right and internal back propagation. So in order to build up to gradient and checking, let's first talk about how to numerically approximate computations of gradients and in the next video, we'll talk about how you can implement gradient checking to make sure the implementation of backdrop is correct. So lets take the function f and replot it here and remember this is f of theta equals theta cubed, and let's again start off to some value of theta. Let's say theta equals 1. Now instead of just nudging theta to the right to get theta plus epsilon, we're going to nudge it to the right and nudge it to the left to get theta minus epsilon, as was theta plus epsilon. So this is 1, this is 1.01, this is 0.99 where, again, epsilon is same as before, it is 0.01. It turns out that rather than taking this little triangle and computing the height over the width, you can get a much better estimate of the gradient if you take this point, f of theta minus epsilon and this point, and you instead compute the height over width of this bigger triangle. So for technical reasons which I won't go into, the height over width of this bigger green triangle gives you a much better approximation to the derivative at theta. And you saw it yourself, taking just this lower triangle in the upper right is as if you have two triangles, right? This one on the upper right and this one on the lower left. And you're kind of taking both of them into account by using this bigger green triangle. So rather than a one sided difference, you're taking a two sided difference. So let's work out

the math. This point here is F of theta plus epsilon. This point here is F of theta minus epsilon. So the height of this big green triangle is f of theta plus epsilon minus f of theta minus epsilon. And then the width, this is 1 epsilon, this is 2 epsilon. So the width of this green triangle is 2 epsilon. So the height of the width is going to be first the height, so that's F of theta plus epsilon minus F of theta minus epsilon divided by the width. So that was 2 epsilon which we write that down here.

2:38
And this should hopefully be close to g of theta. So plug in the values, remember f of theta is theta cubed. So this is theta plus epsilon is 1.01. So I take a cube of that minus 0.99 theta cube of that divided by 2 times 0.01. Feel free to pause the video and practice in the calculator. You should get that this is 3.0001. Whereas from the previous slide, we saw that g of theta, this was 3 theta squared so when theta was 1, so these two values are actually very close to each other. The approximation error is now 0.0001. Whereas on the previous slide, we've taken the one sided of difference just theta + theta + epsilon we had gotten 3.0301 and so the approximation error was 0.03 rather than 0.0001. So this two sided difference way of approximating the derivative you find that this is extremely close to 3. And so this gives you a much greater confidence that g of theta is probably a correct implementation of the derivative of F.

3:58
When you use this method for grading, checking and back propagation, this turns out to run twice as slow as you were to use a one-sided defense. It turns out that in practice I think it's worth it to use this other method because it's just much more accurate. The little bit of optional theory for those of you that are a little bit more familiar of Calculus, it turns out that, and it's okay if you don't get what I'm about to say here. But it turns out that the formal definition of a derivative is for very small values of epsilon is f of theta plus epsilon minus f of theta minus epsilon over 2 epsilon. And the formal definition of derivative is in the limits of exactly that formula on the right as epsilon those as 0. And the definition of unlimited is something that you learned if you took a Calculus class but I won't go into that here. And it turns out that for a non zero value of epsilon, you can show that the error of this approximation is on the order of epsilon squared, and remember epsilon is a very small number. So if epsilon is 0.01 which it is here then epsilon squared is 0.0001. The big O notation means the error is actually some constant times this, but this is actually exactly our approximation error. So the big O constant happens to be 1. Whereas in contrast if we were to use this formula, the other one, then the error is on the order of epsilon. And again, when epsilon is a number less than 1, then epsilon is actually much bigger than epsilon squared which is why this formula here is actually much less accurate approximation than this formula on the left. Which is why when doing gradient checking, we rather use this two-sided difference when you compute f of theta plus epsilon minus f of theta minus epsilon and then divide by 2 epsilon rather than just one sided difference which is less accurate.

5:53
If you didn't understand my last two comments, all of these things are on here. Don't worry about it. That's really more for those of you that are a bit more familiar with Calculus, and with numerical approximations. But the takeaway is that this two-sided difference formula is much more accurate. And so that's what we're going to use when we do gradient checking in the next video.

6:13
So you've seen how by taking a two sided difference, you can numerically verify whether or not a function g, g of theta that someone else gives you is a correct implementation of the derivative of a function f. Let's now see how we can use this to verify whether or not your back propagation implementation is correct or if there might be a bug in there that you need to go and tease out

Downloads

Lecture Video	mp4
Subtitles (English)	WebVTT
Transcript (English)	txt
Lecture Notes	pptx

Would you like to help us translate the transcript and subtitles into additional languages?