Computation graph

(SPEECH)
You've

(DESCRIPTION)
Text, deep learning dot AI. Basics of neural network programming. Computation graph

(SPEECH)
heard me say that the computations of a neural network are organized in terms of a forward pass or a forward propagation step, in which we compute the output of the neural network, followed by a backward pass or back propagation step, which we use to compute gradients or compute derivatives.

The computation graph explains why it is organized this way.

In this video, we'll go through an example.

In order to illustrate the computation graph, let's use a simpler example than logistic regression or a full blown neural network.

Let's say that we're trying to compute a function, J, which is a function of three variables a, b, and c and let's say that function is 3(a+bc).

Computing this function actually has three distinct steps.

The first is you need to compute what is bc and let's say we store that in the variable call u.

So u=bc and then you my compute V=a *u.

So

(DESCRIPTION)
Text, the audio for v's formula is incorrect. v equals A plus u is correct

(SPEECH)
let's say this is V. And then finally, your output J is 3V.

(DESCRIPTION)
Using brackets to break the formula for J into three substeps, where the innermost computation for u is done first, then the computation for v depending on u, then the computation for J depending on v

(SPEECH)
So this is your final function J that you're trying to compute.

We can take these three steps and draw them in a computation graph as follows.

Let's say, I draw your three variables a, b, and c here.

So

(DESCRIPTION)
Three rows, one for each input variable, on the left

(SPEECH)
the first thing we did was compute u=bc.

So I'm going to put a rectangular box around that.

And so the input to that are b and c. And

(DESCRIPTION)
Drawing arrows from b and c rightward to the boxed formula for u which depends on their values

(SPEECH)
then, you might have V=a+u.

(DESCRIPTION)
Another boxed formula

(SPEECH)
So the inputs to that are V. So the inputs to that are u with just computed together with a.

(DESCRIPTION)
Drawing arrows from A and the first boxed formula, to the second boxed formula

(SPEECH)
And then finally, we have J=3V.

(DESCRIPTION)
Third boxed formula. Drawing just one arrow from the second boxed formula to the third

(SPEECH)
So as a concrete example, if a=5, b=3 and c=2 then u=bc would be six because a+u would be 5+6 is 11, J is three times that, so J=33.

And

(DESCRIPTION)
Adding these values to the input variables, then successively computing the values of the boxed formulas which depend on those values

(SPEECH)
indeed, hopefully you can verify that this is three times five plus three times two.

And if you expand that out, you actually get 33 as the value of J.

So, the computation graph comes in handy when there is some distinguished or some special output variable, such as J in this case, that you want to optimize.

And in the case of a logistic regression, J is of course the cos function that we're trying to minimize.

And what we're seeing in this little example is that, through a left-to-right pass, you can compute the value of J.

And

(DESCRIPTION)
Drawing arrows in the opposite direction, right to left, from the output box to the input variables

(SPEECH)
what we'll see in the next couple of slides is that in order to compute derivatives, there'll be a right-to-left pass like this, kind of going in the opposite direction as the blue arrows.

That would be most natural for computing the derivatives.

So to recap, the computation graph organizes a computation with this blue arrow, left-to-right computation.

Let's refer to the next video how you can do the backward red arrow right-to-left computation of the derivatives.

Let's go on to the next video.