Neural Networks Overview

(SPEECH)
Welcome

(DESCRIPTION)
Text, deep learning dot AI. One hidden layer neural network. Neural networks overview

(SPEECH)
back. In this week, you learned to implement a neural network.

Before diving into the technical details, I want in this video, to give you a quick overview of what you'll be seeing in this week's videos.

So, if you don't follow all the details in this video, don't worry about it, we'll delve into the technical details in the next few videos.

But for now, let's give a quick overview of how you implement in your network.

Last

(DESCRIPTION)
Text, what is a neural network? Diagram of three input features, x 1, x 2, and x 3, passing into a node, which produces output y hat

(SPEECH)
week, we had talked about logistic regression, and we saw how this model corresponds to the following computation draft, where you didn't put the features x and parameters w and b that allows you to compute z which is then used to computes a, and we were using a interchangeably with this output y hat and then you can compute the loss function, L. A

(DESCRIPTION)
Diagram with the same structure, where the input features are x, w, and b, the node is a computation with the formula z equals w transpose x plus b, and that node sends its output to another node which computes A equals sigmoid of z

(SPEECH)
neural network looks like this.

As

(DESCRIPTION)
Another network diagram with a more intricate structure. Still taking three input features, but there are three nodes each taking all three features as inputs, and each of those those three nodes sends its output to a central node which in turn outputs y hat

(SPEECH)
I'd already previously alluded, you can form a neural network by stacking together a lot of little sigmoid units.

(DESCRIPTION)
Indicating the central node of the less complicated first diagram

(SPEECH)
Whereas previously, this node corresponds to two steps to calculations.

The

(DESCRIPTION)
The first step of the computation graph computing z from x, w, and b, and the second computing A from sigmoid of z

(SPEECH)
first is compute the z-value, second is it computes this a value.

In this neural network, this stack of notes will correspond to a z-like calculation like this, as well as, an a-like calculation like that.

Then, that node will correspond to another z and another a like calculation.

So the notation which we will introduce later will look like this.

First, we'll inputs the features, x, together with some parameters w and b, and this will allow you to compute z one.

So,

(DESCRIPTION)
z superscript square bracket 1, equals, W superscript square bracket 1 times x, plus, b superscript square bracket 1

(SPEECH)
new notation that we'll introduce is that we'll use superscript square bracket one to refer to quantities associated with this stack of nodes, it's called a layer.

Then later, we'll use superscript square bracket two to refer to quantities associated with that node.

That's called another layer of the neural network.

The superscript square brackets, like we have here, are not to be confused with the superscript round brackets which we use to refer to individual training examples.

So, whereas x superscript round bracket I refer to the ith training example, superscript square bracket one and two refer to these different layers; layer one and layer two in this neural network.

But so going on, after computing z_1 similar to logistic regression, there'll be a computation to compute a_1, and that's just sigmoid of z_1, and then you compute z_2 using another linear equation

(DESCRIPTION)
z 2, equals, W 2 times a 1, plus b 2

(SPEECH)
and then compute a_2.

(DESCRIPTION)
a 2 equals sigmoid of z 2

(SPEECH)
A_2 is the final output of the neural network and will also be used interchangeably with y-hat.

(DESCRIPTION)
Also produces loss function output computation, L of a 2 and y

(SPEECH)
So, I know that was a lot of details but the key intuition to take away is that whereas for logistic regression, we had this z followed by a calculation.

In this neural network, here we just do it multiple times, as a z followed by a calculation, and a z followed by a calculation, and then you finally compute the loss at the end.

You remember that for logistic regression, we had this backward calculation in order to compute derivatives or as you're computing your d a, d z and so on.

So,

(DESCRIPTION)
Starting with loss function at endpoint of logistic regression chart, drawing arrows backwards through the steps to arrive at input feature nodes

(SPEECH)
in the same way, a neural network will end up doing a backward calculation that looks like this in which you end up computing da_2, dz_2, that allows you to compute dw_2, db_2, and so on.

(DESCRIPTION)
Starting with loss function at endpoint of neural network, drawing arrows backwards through the steps to arrive at input feature nodes, written in superscript notation

(SPEECH)
This right to left backward calculation that is denoting with the red arrows.

So, that gives you a quick overview of what a neural network looks like.

It's basically taken logistic regression and repeating it twice.

I know there was a lot of new notation laws, new details, don't worry about saving them, follow everything, we'll go into the details most probably in the next few videos.

So, let's go on to the next video.

We'll start to talk about the neural network representation.